



**HAL**  
open science

## Facial Micro-Expression Analysis

Jingting Li

► **To cite this version:**

Jingting Li. Facial Micro-Expression Analysis. Computer Vision and Pattern Recognition [cs.CV]. CentraleSupélec, 2019. English. NNT: 2019CSUP0007 . tel-02877766

**HAL Id: tel-02877766**

**<https://theses.hal.science/tel-02877766v1>**

Submitted on 22 Jun 2020

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# THESE DE DOCTORAT DE

CENTRALESUPELEC

COMUE UNIVERSITE BRETAGNE LOIRE

ECOLE DOCTORALE N° 601

*Mathématiques et Sciences et Technologies  
de l'Information et de la Communication*

Spécialité : *Signal, Image, Vision*

Par

« **Jingting LI** »

« **Facial Micro-Expression Analysis** »

Thèse présentée et soutenue à « Rennes », le « 02/12/2019 »

Unité de recherche : IETR

Thèse N° : 2019CSUP0007

## Rapporteurs avant soutenance :

Olivier ALATA	Professeur, Université Jean Monnet, Saint-Etienne
Fan YANG-SONG	Professeur, Université de Bourgogne, Dijon

## Composition du Jury :

Président / Rapporteur Olivier ALATA	Professeur, Université Jean Monnet, Saint-Etienne
Rapporteuse Fan YANG-SONG	Professeur, Université de Bourgogne, Dijon
Examineurs Catherine SOLADIE	Maître de Conférence, CentraleSupélec, Rennes
Pascal BOURDON	Maître de Conférence, Université de Poitiers, Poitiers
Wassim HAMIDOUCHE	Maître de Conférence, INSA de Rennes, Rennes
Directeur de thèse Renaud SEGUIER	Professeur HDR, CentraleSupélec, Rennes

**Titre :** L'Analyse de Micro-Expression Faciale

**Mots clés :** Micro-expression, Détection, Motif temporel et local, Augmentation des données, Modèle d'Hammerstein

**Résumé :** Les micro-expressions (MEs) sont porteuses d'informations non verbales spécifiques. Cependant, en raison de leur nature locale et brève, il est difficile de les détecter. Dans cette thèse, nous proposons une méthode de détection par reconnaissance d'un motif local et temporel de mouvement du visage. Ce motif a une forme spécifique (S-pattern) lorsque la ME apparaît. Ainsi, à l'aide de SVM, nous distinguons les MEs des autres mouvements faciaux. Nous proposons également une fusion spatiale et temporelle afin d'améliorer la distinction entre les MEs (locaux) et les mouvements de la tête (globaux). Cependant, l'apprentissage des S-patterns est limité par le petit nombre de bases de données de ME et par le faible volume d'échantillons de ME. Les modèles de Hammerstein (HM) est une bonne approximation des mouvements musculaires.

En approximant chaque S-pattern par un HM, nous pouvons filtrer les S-patterns réels et générer de nouveaux S-patterns similaires. Ainsi, nous effectuons une augmentation et une fiabilisation des S-patterns pour l'apprentissage et améliorons ainsi la capacité de différencier les MEs d'autres mouvements. Lors du premier challenge de détection de MEs, nous avons participé à la création d'une nouvelle méthode d'évaluation des résultats. Cela a aussi été l'occasion d'appliquer notre méthode à longues vidéos. Nous avons fourni le résultat de base au challenge. Les expérimentations sont effectuées sur CASME I, CASME II, SAMM et CAS(ME)<sup>2</sup>. Les résultats montrent que notre méthode proposée surpasse la méthode la plus populaire en termes de F1-score. L'ajout du processus de fusion et de l'augmentation des données améliore encore les performances de détection.

**Title :** Facial Micro-expression Analysis

**Keywords:** Micro-expression, Spotting, Local temporal pattern, Data augmentation, Hammerstein model

**Abstract:** The Micro-expressions (MEs) are very important nonverbal communication clues. However, due to their local and short nature, spotting them is challenging. In this thesis, we address this problem by using a dedicated local and temporal pattern (LTP) of facial movement. This pattern has a specific shape (S-pattern) when ME are displayed. Thus, by using a classical classification algorithm (SVM), MEs are distinguished from other facial movements. We also propose a global final fusion analysis on the whole face to improve the distinction between ME (local) and head (global) movements. However, the learning of S-patterns is limited by the small number of ME databases and the low volume of ME samples. Hammerstein models (HMs) are known to be a good approximation of muscle movements.

By approximating each S-pattern with a HM, we can both filter outliers and generate new similar S-patterns. By this way, we perform a data augmentation for S-pattern training dataset and improve the ability to differentiate MEs from other facial movements. In the first ME spotting challenge of MEGC2019, we took part in the building of the new result evaluation method. In addition, we applied our method to spotting ME in long videos and provided the baseline result for the challenge. The spotting results, performed on CASME I and CASME II, SAMM and CAS(ME)<sup>2</sup>, show that our proposed LTP outperforms the most popular spotting method in terms of F1-score. Adding the fusion process and data augmentation improve even more the spotting performance.

# Acknowledgement

There are many people that have earned my gratitude for their contribution to my PhD study.

Firstly, I would like to express my sincere gratitude to my advisors Prof. Renaud Séguier and Dr. Catherine Soladié for the continuous support of my PhD study and related research, for their patience, motivation, and immense knowledge. The guidance of Dr. Catherine Soladié helped me in all the time of research and writing of this thesis. She is the best!

Besides my advisors, I would like to thank the rest of my thesis committee: Prof. Olivier Alata, Prof. Fan Yang-Song, Dr. Pascal Bourdon, and Dr. Wassim Hamidouche for their great support and invaluable advice, which encourages me to widen my research from various perspectives.

I would also like to thank my lab mates that include Siwei Wang, Sarra Zaied, Sen Yan, Nam Duong Duong, Dai Xiang, Amanda Abreu, Raphael Weber, Salah Eddine Kabbour, Corentin Guezenoc, Adrien Llave, Eloïse Berson, and Lara Younes for making my experience in our small but lovely campus exciting and fun in the last three years.

I am also grateful to the following university staffs: Karine Bernard, Bernard Jouga, Cecile Dubois; Catherine Piednoir and Grégory Cadeau for their unfailing support and assistance.

Thanks are also due to the China Scholar Council (CSC) and ANR Reflet for their financial support that I otherwise would not have been able to develop my scientific discoveries.

Last but not least, I would like to express my deepest gratitude to my big and warm family: my grandparents, my parents, my sister, and my baby nieces, also to my dear friends: Chunmei Xie and Yi Hui. This dissertation would not have been possible without



their warm love, continued patience, and endless support.

# Résumé Français

## Chapitre 1: Introduction

L'expression faciale est l'un des indicateurs externes les plus importants pour connaître l'émotion et le statut psychologique d'une personne [12]. Parmi les expressions faciales, les micro-expressions (MEs) [28] sont des expressions locales et brèves qui apparaissent involontairement, notamment dans le cas de forte pression émotionnelle. Leurs durées varient de 1/25 à 1/5 de seconde [28]. Leur caractère involontaire permet souvent d'affirmer qu'elles représentent des émotions véritables d'une personne [28]. La détection de MEs a des applications nombreuses notamment dans le domaine de la sécurité nationale [26], des soins médicaux [30], des études sur la psychologie politique [108] et la psychologie de l'éducation [17].

L'existence de MEs a d'abord été découverte par Haggard et Isaacs en 1966 [39] puis Ekman et Friesen [28] l'ont nommée en 1969. Plusieurs années plus tard, ils ont développé un outil pour former les personnes à la détection de micro-expressions (METT) [25]. Pourtant, le taux de reconnaissance global pour les 6 émotions de base à l'œil nu est inférieur à 50%, même par un expert formé [31].

Pour coder les MEs, le système de codage d'actions faciales (FACS) [27] est souvent utilisé. Il a été créé pour analyser la relation entre la déformation du muscle facial et l'expression émotionnelle. Les unités d'action (AUs) sont les composantes faciales du FACS, qui représentent le mouvement musculaire local. L'étiquette de l'AU sur le visage permet d'identifier la ou les régions où la ME se produit. En conséquence, le système FACS peut aider à annoter l'apparence et la dynamique d'une ME dans une vidéo.

Depuis les années 2000, la recherche sur la détection et la reconnaissance automatique

de micro-expressions (micro-expression spotting and recognition, MESR) s'est développée. La figure 0-1 indique l'évolution du nombre d'articles de recherche MESR. Le nombre des articles reste faible et les résultats ne sont pas encore très satisfaisants du fait de la nature même des MEs (micro) ainsi que du nombre limité de bases de données (BDDs) publiques de MEs. Cependant, il y a eu de plus en plus d'études émergentes ces dernières années. Nous pouvons remarquer qu'il y a beaucoup plus de papiers qui concernent la reconnaissance de ME que la détection. Le taux de reconnaissance commence à être satisfaisant, par exemple 86.35% de précision pour 5 classes dans [20]. En revanche, les résultats de la détection de ME sont loin d'être bons, certainement à cause de la nature de ME et également du nombre limité de bases de données publiques de ME. En effet, la plupart

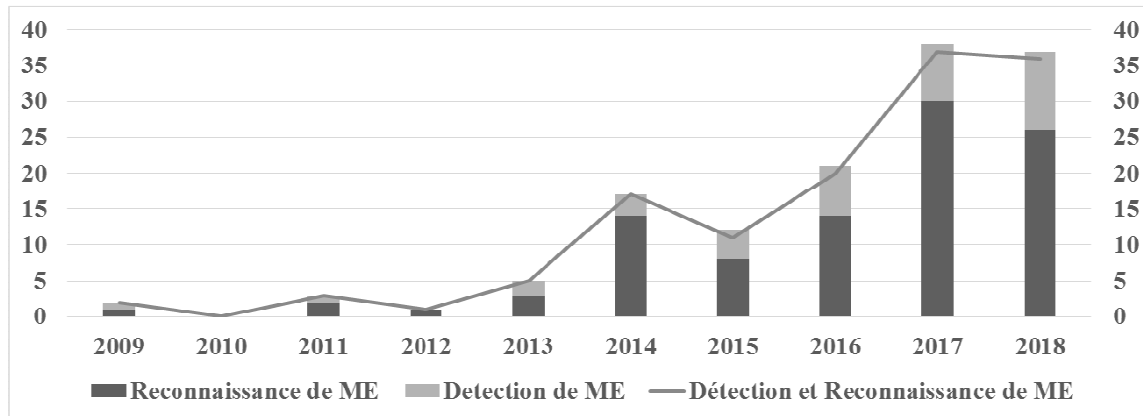


Figure 0-1: Tendence de la recherche MESR. Le nombre d'articles sur MESR augmente d'année en année, principalement dans le domaine de la reconnaissance de l'ME (colonne du bas). La recherche sur la détection de ME n'a pas encore suffisamment attiré l'attention (colonne en haut).

des méthodes de reconnaissance de ME supposent connues les images d'onset et d'offset des MEs. Mais trouver les images de départ (onset) et des images de fin (offset), en particulier lorsque les ME sont des mouvements micro-faciaux spontanés dans une séquence vidéo complète, reste un énorme défi. Les résultats de détection des méthodes proposées actuellement ne sont pas assez précis. Par exemple, dans la tâche de détection du deuxième grand challenge de micro-expression faciale (MEGC2019) [64], le F1-score de base est inférieure à 0,05. Même lorsque les ME sont produites dans un environnement strictement contrôlé, les faux positifs sont très nombreux en raison des mouvements de la tête ou du

clignement des yeux.

L'analyse de micro-expression est limitée par le petit nombre de données. Cela est dû à la taille des bases de données de ME. Par exemple, la plus grande base de données de micro-expressions spontanées ne contient que 255 échantillons vidéo. Cette situation limite grandement l'utilisation de l'apprentissage automatique pour la détection de ME.

Un autre problème non résolu en ce qui concerne la détection de micro-expressions est que les métriques utilisées pour analyser le résultat dans différents papiers sont diverses. Les précisions sont étudiées par image, par intervalle ou par vidéo, tandis que la mesure peut être TPR, ROC, ACC ou encore d'autres mesures. Les métriques utilisées sont souvent choisies en fonction de la méthode proposée. Comme les méthodes ne sont pas les mêmes, chaque papier utilise des métriques différentes des autres papiers. Il est alors difficile de comparer les résultats.

Dans cette thèse, nous explorons un système automatique permettant de détecter des MEs. Un tel système doit être capable de :

- détecter des frames de micro-expression dans des séquences vidéo
- séparer les mouvements relatifs aux MEs des mouvements de la tête ou du clignement des yeux.
- détecter la région où la ME se produit.
- traiter le problème du petit nombre de données.

Nous avons quatre contributions dans cette thèse. La principale consiste à proposer un motif temporel local (local temporal pattern, LTP) dédié, qui extrait les informations pertinentes pour la détection de ME [63]. Une ME étant un mouvement bref et local, le modèle de mouvement est analysé à partir des régions de visage locales. Les emplacements sont les petites régions d'intérêt (ROI) où des ME peuvent se produire. Lorsqu'il y a une ME, la texture (c'est-à-dire la valeur du niveau de gris) de la ROI change. Nous calculons donc ce changement, qui correspond à la distance entre le niveau de gris de 2 frames de la même ROI. L'une des originalités de cette approche réside dans l'utilisation d'un motif temporel sur ces ROIs. La durée est celle de ME (300ms). La courbe de la figure 0-2

représente la variation temporelle locale de la texture (au niveau des gris) lorsqu'une ME se produit dans la ROI. Nous pouvons remarquer qu'il forme un motif en S (S-pattern) depuis le début jusqu'à l'apex de la ME. Ce S-pattern apparaît chaque fois qu'une micro-expression se produit dans une région et il est indépendant de la ROI et du sujet, ce qui la rend pertinente pour la détection de ME. Plus précisément, ce motif de LTP est calculé sur un intervalle de la durée d'une ME (300 ms) et le motif est une liste de distances entre les textures de niveau de gris du premier frame et du  $k^{ième}$  frame de l'intervalle. Afin de conserver la variation la plus significative, une analyse en composantes principales (ACP) est d'abord effectuée (pour la ROI correspondant) sur l'ensemble de la vidéo.

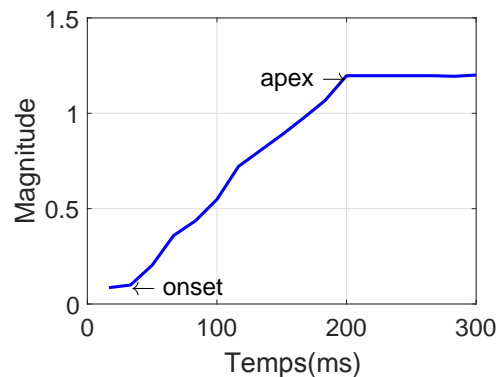


Figure 0-2: Exemple de motif temporel local (LTP) au cours d'une ME située dans la région du sourcil droit sur une période de 300 ms (durée moyenne de ME). Ce LTP représente l'évolution de la texture (niveau de gris) d'une ROI pendant la ME. Il forme un motif en S (S-pattern). La courbe atteint son sommet au bout de 150 ms environ, puis reste stable ou diminue légèrement. Ce motif est spécifique aux mouvements ME et est appelé motif en S (S-pattern) en raison de la forme de la courbe. (Vidéo: Sub01\_EP01\_5 of CASME I)

La deuxième contribution concerne l'élimination des mouvements liés au mouvement de la tête ou au clignement des yeux. La particularité de cette approche est la combinaison de traitements locaux et globaux. La détection de LTP est effectuée localement dans les ROIs. Un système de fusion sur tout le visage sépare ensuite les ME des autres mouvements du visage. Enfin, en fonction des S-patterns détectés, il est possible de déterminer l'indice temporel du début de la ME. Notez que, comme le LTP est local, nous avons également les informations sur le lieu où la ME se produit.

Une autre contribution principale concerne l'amélioration de la détection à la fois en filtrant les mauvais S-patterns et en augmentant les données. En effet, les échantillons dans

les bases de données de ME existantes sont faibles, ce qui limite grandement l'amélioration des performances de détection de ME. De plus, les bases de données ne sont pas étiquetées avec l'emplacement précis des MEs (la partie gauche ou droite du visage n'est pas mentionnée par exemple). Cela conduit à des annotations erronées de LTPs. L'originalité est d'utiliser le modèle d'Hammerstein (Hammerstein model, HM), qui est connu pour approximer efficacement les mouvements musculaires. Chaque S-pattern peut être approximé par un HM avec deux paramètres. En utilisant la distribution de ces deux paramètres, nous pouvons filtrer les mauvais motifs et générer d'autres S-patterns similaires. L'utilisation de ces modèles contribue à l'extension et à la fiabilité des échantillons de données pour l'apprentissage de S-patterns. L'entraînement est ensuite effectuée sur des motifs réels et synthétiques.

La dernière contribution est la tâche de détection du deuxième grand challenge de Micro-Expression (MEGC2019). Nous avons fourni la méthode de base et le résultat en détectant les ME sur de longues séquences vidéo. De plus, pour assurer la cohérence de la méthode d'évaluation des résultats, nous fournissons un ensemble de nouvelles mesures de performance. Il a été utilisé comme guide pour l'évaluation des résultats de ME détection dans MEGC2019.

Le document est organisé comme suit: le chapitre 2 mène une revue sur l'état de l'art de la détection et de la reconnaissance automatique de micro-expressions faciales (MESR). Le chapitre 3 décrit notre méthode en utilisant le motif temporel local (LTP) pour la détection de ME. Le chapitre 4 propose d'augmenter la taille de la base de données de S-pattern pour l'étape d'apprentissage de la classification locale. En effet, le petit volume des bases de données de S-pattern limite les performances de notre méthode. Le chapitre 5 présente les résultats expérimentaux de notre méthode. Le chapitre 6 résume nos contributions à cette thèse et présente les perspectives des travaux futurs.

## **Chapitre 2: État de l'art**

Dans ce chapitre, nous menons une revue sur l'état de l'art de la reconnaissance et de la détection automatique de micro-expressions faciales (MESR). Tout d'abord, nous intro-

duisons une analyse systématique des bases de données de micro-expression<sup>(1)</sup>. Deuxièmement, une analyse des méthodes d'évaluation des résultats et des métriques pour l'analyse des micro-expressions est présentée<sup>(2)</sup>. Troisièmement, toutes les méthodes de détection publiées sont analysées avec leurs avantages et leurs inconvénients<sup>(3)</sup>. Ensuite, le schéma de la détection et de la reconnaissance de ME est discuté<sup>(4)</sup>. Enfin, nous donnons notre point de vue sur ce domaine de recherche et les contributions de notre méthode<sup>(5)</sup>.

1). Les bases de données sont analysées selon 13 caractéristiques regroupées en quatre catégories (population, matériel, protocole expérimental et annotation). Ces caractéristiques fournissent une référence non seulement pour le choix d'une base de données à des fins d'analyse spéciales, mais également pour la construction future de bases de données.

2). Concernant les méthodes d'évaluation des résultats et métriques, nous proposons une revue complète. Nous nous concentrons sur les méthodes d'évaluation des résultats et les métriques pour la reconnaissance et la détection des ME, respectivement. Pour la reconnaissance, outre un résumé quantitatif pour chaque métrique, nous avons également examiné et discuté le nombre de classes de reconnaissance. En effet, le nombre de classes impacte grandement le résultat de reconnaissance. En ce qui concerne la détection de ME, les métriques sont introduites en fonction des différentes méthodes de détection. Cette section présente également une discussion sur la standardisation des métriques.

3). Nos recherches portent sur la détection de micro-expressions. La revue des travaux connexes indique les avantages et les désavantages des méthodes de détection actuelles. Comme une micro-expression est une expression faciale involontaire, nous nous concentrons sur les méthodes de détection de ME développées à partir de bases de données spontanées ou à "in-the-wild". Nous comparons d'abord les méthodes en fonction de leurs algorithmes, puis nous étudions les descripteurs pour la détection de micro-expressions. Sur la base de ces analyses, nous concluons cette section et éclairons l'orientation de nos recherches.

4). Dans cette section, nous discutons des schémas de détection et de reconnaissance de micro-expression. Comme l'analyse de micro-expression automatique est censée être appliquée dans la vie réelle, il convient de prendre en compte un processus complet dans lequel la séquence vidéo constitue l'entrée et la classe de l'émotion est la sortie. De

notre point de vue, il existe deux types de schémas. L'un consiste à traiter la non-micro-expression comme une classe émotionnelle, puis à appliquer une méthode de reconnaissance pour classifier les échantillons en différentes classes émotionnelles. L'autre consiste tout d'abord à détecter les séquences de micro-expression dans une longue vidéo, puis à identifier le type d'émotion de cette séquence ME par des méthodes de reconnaissance.

5). En conclusion, la recherche sur la détection de micro-expressions est importante pour les applications dans la vie réelle. Nous étudions les caractéristiques locales et temporelles pour la détection de micro-expressions. Une fusion tardive du local au global est appliquée pour améliorer la capacité de distinguer la micro-expression et les autres mouvements faciaux. De plus, nous explorons les méthodes d'augmentation des données afin de résoudre le problème du petit nombre d'échantillons de micro-expression.

## **Chapitre 3: Motif local et temporel pour la détection de micro-expressions**

Dans ce chapitre, nous proposons notre méthode pour détecter les micro-expressions. Il se concentre particulièrement sur deux contributions: 1). la définition d'un nouveau descripteur pertinent pour la détection de micro-expressions: le motif temporel local (local temporal pattern, LTP); 2). la fusion spatiale et temporelle tardive, réalisée pour obtenir le résultat final de détection de micro-expressions. La Figure 0-3 affiche le processus global. La méthode proposée comprend trois parties: un pré-traitement pour détecter avec précision les points caractéristiques du visage et extraire les régions d'intérêt (ROIs), puis le calcul du motif temporel local (LTP) sur ces ROIs et finalement la détection de micro-expressions. Les trois premières sections de ce chapitre présentent les sous-étapes de notre méthode appliquées dans de courtes vidéos. Notre méthode est ensuite adaptée aux situations de longues vidéos. Enfin, nous concluons le chapitre et soulignons les exigences pour améliorer les performances de notre méthode.



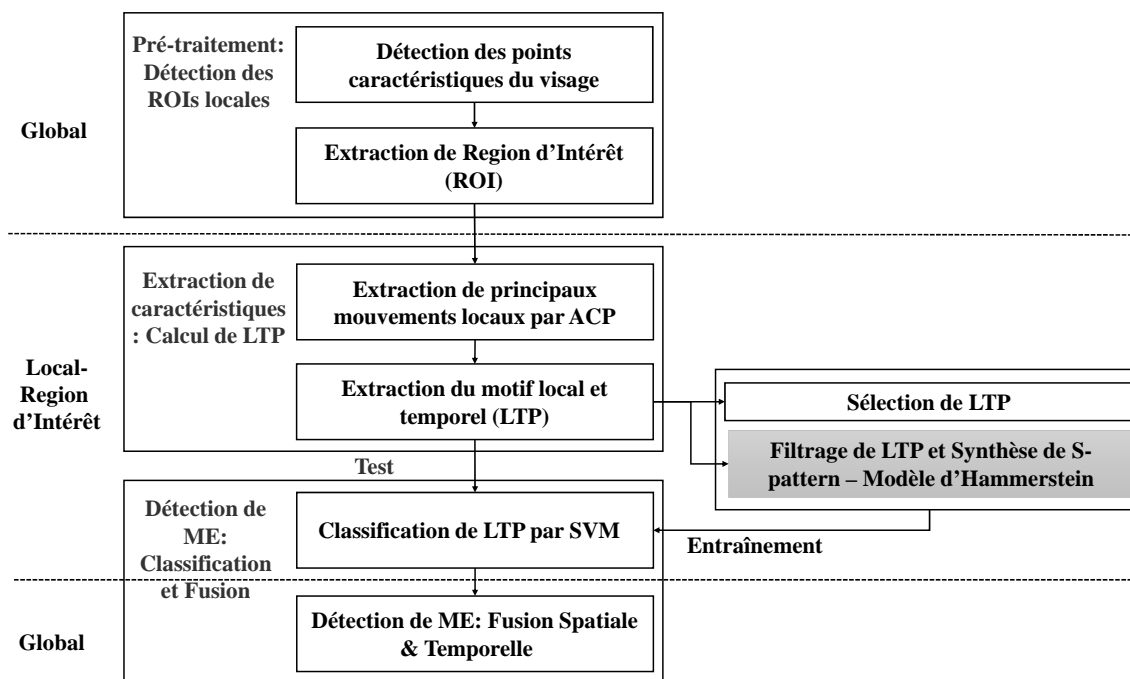


Figure 0-3: Vue d'ensemble de notre méthode. La méthode proposée comporte trois étapes de traitement: pré-traitement, extraction de caractéristiques et détection de micro-expressions. Nous mélangeons des processus globaux (tout le visage) et locaux (les ROIs). La sous-étape d'extraction des caractéristiques et la première sous-étape de la détection de la micro-expression sont effectuées sur des régions d'intérêt (ROIs) pertinentes, tandis que les autres étapes sont réalisées sur tout le visage (global). Les LTPs, y compris les S-patterns, sont ensuite utilisés comme échantillons d'apprentissage pour construire le modèle d'apprentissage automatique (SVM) en vue de la classification. En particulier, une fusion spatiale et temporelle finale est réalisée pour éliminer les faux positifs tels que les clignements des yeux. L'une des spécificités du processus réside dans l'utilisation de motifs temporels locaux (LTP), pertinents pour la détection de micro-expressions: les micro-expressions sont des mouvements brefs et locaux.

## Pré-traitement

Comme la micro-expression est un mouvement facial local, l'analyse sur une région locale permet d'extraire des caractéristiques plus pertinentes. Le pré-traitement est effectué sur le visage pour déterminer les régions d'intérêt locales (ROI). Ce processus comporte deux étapes: les points caractéristiques du visage sont d'abord détectés, puis les points liés aux micro-expressions sont choisis pour extraire les régions d'intérêt (ROIs).

## Extraction de caractéristiques: calcul du motif temporel local

L'objectif de cette section est d'extraire un nouveau descripteur pertinent pour la détection des micro-expressions: les motifs temporels locaux (LTPs). Les micro-expressions étant des mouvements locaux brefs, les LTPs visent à extraire les informations locales sur la distorsion de la texture dans une fenêtre temporelle de la taille d'une micro-expression (300 ms).

Les LTPs sont calculés pour chaque image et chaque ROI. Ils sont basés sur le changement de texture du niveau de gris du ROI. Pour détecter la distorsion principale de la texture de niveau de gris d'une ROI en fonction du temps, nous utilisons une ACP sur toute la séquence de ROI. La figure 0-4 illustre ce traitement sur l'une des séquences de ROI.

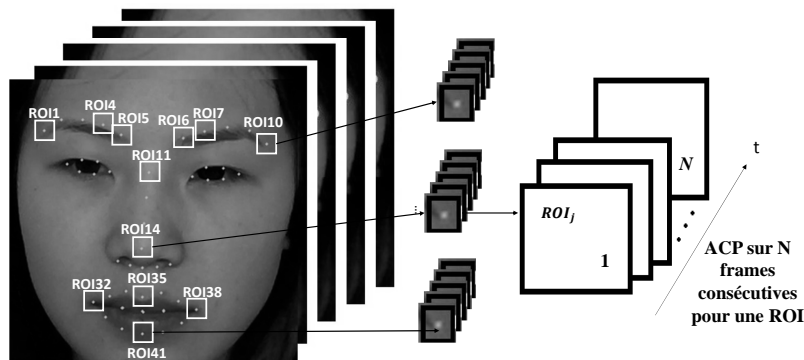


Figure 0-4: ACP sur l'axe des temps par ROI. Une séquence vidéo locale  $ROI_j$  avec  $N$  images (durée de la vidéo  $\leq 3$ s) est traitée par l'ACP sur l'axe des temps. Les premières composantes de l'ACP conservent le mouvement principal de la texture de niveau de gris sur cette ROI pendant cette durée ( $N$  images). L'échantillon vidéo provient de CASME I (©Xiaolan Fu)

Après avoir obtenu la distribution des frames de la séquence de ROI par la réduction des

dimensions (ACP), un 2D point dans la distribution représente une ROI frame. En calculant de distance entre les points et faisant une normalisation, les LTP sont obtenus. Enfin, nous démontrons que le S-pattern est unique pour toutes les micro-expressions.

## **Détection de micro-expressions: classification et fusion**

La détection de la micro-expression est traité en deux étapes: une classification locale de LTP et une fusion spatio-temporelle (voir la figure 0-3). En ce qui concerne la classification de LTP, comme montré dans la figure 0-2, le S-pattern représente le mouvement de la micro-expression. Et il est identique pour toutes sortes d'émotions. Donc, les LTPs sont d'abord sélectionnés pour un entraînement efficace en apprentissage automatique. Ensuite, les LTPs sont classés comme S-patterns ou non-S-patterns à l'étape de test. Enfin, une analyse de fusion du local au global est effectuée pour obtenir un résultat global pour chaque image. Son objectif est à la fois d'éliminer le mouvement global de la tête et de fusionner les résultats positifs locaux de différentes ROIs appartenant à la même micro-expression globale.

Pour l'étape d'entraînement, les LTPs doivent être séparés en 2 groupes: le S-pattern (micro-expression) et le non-S-pattern (autre mouvement du visage). Dans les bases de données publiques, les séquences de micro-expression sont annotées avec les informations d'onset et les AUs. Ainsi, nous devons pré-traiter les étiquettes des bases de données pour obtenir la vérité terrain pour notre entraînement. Comme montré dans la figure 0-5, cette labellisation s'effectue en 3 étapes: une annotation temporelle, une sélection locale (sélection AU par ROI) et une sélection liée à la forme de LTP.

Une fois que l'annotation (S-pattern et non-S-pattern) est effectuée, une classification supervisée (SVM) est utilisé. Les résultats de la classification sont générés par LOSub-OCV (validation croisée de leave-one-class-out). Les ROI avec S-pattern sont reconnus, indiquant qu'un mouvement similaire à la micro-expression se produit dans cette région.

Cette classification permet d'identifier les motif locaux correspondant à une ME. Pourtant, le résultat d'identification de ME doit être global. Ainsi, nous effectuons une fusion spatiale et temporelle. Le processus comprend trois étapes: la qualification locale, la fusion

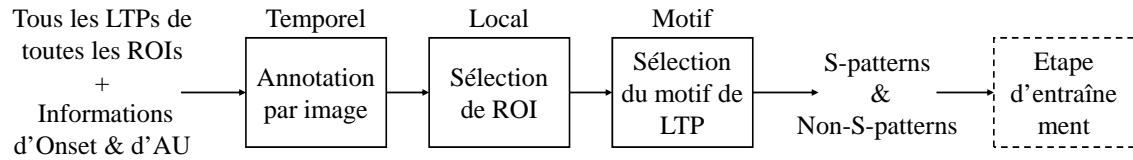


Figure 0-5: Sélection de LTP pour l'étape d'entraînement. Tous les LTPs sont classés en 2 classes: S-patterns et non-S-patterns. Les LTPs passent par 3 étapes pour l'annotation: l'annotation par image, la sélection de ROI à partir de l'AU et la sélection du motif de LTP. Les S-patterns annotés et les non-S-patterns sont ensuite transmis à l'étape d'apprentissage du classifieur SVM.

spatiale et le lissage des frames.

## Détecter les micro-expressions dans les longues vidéos

Les travaux précédents portaient sur la détection de micro-expressions dans de courtes vidéos (moins de 2 secondes). Cependant, dans la vie réelle, les vidéos pour l'analyse de micro-expression sont beaucoup plus longues. Par conséquent, il est nécessaire de développer et de tester notre méthode dans la situation de longue durée. Il existe deux bases de données de micro-expressions spontanées contenant de longues vidéos: SAMM [19] et CAS (ME)<sup>2</sup> [104]. Toutes les expérimentations dans les longues vidéos sont effectuées sur ces deux bases de données.

Cette section présente la modification de notre méthode pour les applications en vidéos longues. Deux étapes sont impactées: le pré-traitement et la détection de ME.

## Chapitre 4: Augmentation des données à l'aide du modèle d'Hammerstein

Comme présenté au chapitre précédent, notre méthode se base sur du machine learning. Elle est donc limitée par la quantité d'échantillons de micro-expression (ME). En effet, il n'y a pas beaucoup de bases de données avec des MEs étiquetées, donc pas beaucoup de images de ME étiquetés. En outre, la quantité de images non ME est supérieure à celle des images ME, c'est-à-dire que la taille de la base de données de S-pattern n'est pas assez

grande. En synthétisant les caractéristiques de micro-expressions (S-pattern), le volume de données d'apprentissage peut être étendu.

Dans ce chapitre, afin d'améliorer les performances de notre méthode, nous proposons d'augmenter la taille de la base de données de S-pattern pour la phase d'entraînement de la classification locale. Pour ce faire, nous utilisons le modèle de Hammerstein (HM), qui est connu pour simuler la dynamique du mouvement musculaire. La figure 0-3 illustre la modification de l'ensemble du processus: la sélection du modèle LTP est remplacée par le filtrage de LTP et la synthèse de S-pattern. Plus précisément, les annotations et la sélection d'AU de la partie "sélection de LTP" dans la figure 0-5 sont conservées. Ensuite, nous remplaçons la partie "sélection du motif de LTP" de la troisième étape par un filtrage du LTP et une synthèse du S-pattern par le modèle d'Hammerstein. Le schéma de ce nouveau sous-processus est illustré à la figure 0-6. Les S-patterns issus de l'annotation des étiquettes et de la sélection AU (S-patterns<sub>O</sub>) sont d'abord modélisés par l'identification du modèle d'Hammerstein. Ils sont ensuite filtrés, et les S-patterns restants (S-patterns<sub>OF</sub>), servent de base à la synthèse de davantage de S-patterns (S-patterns<sub>ST</sub>).

## Hammerstein Model

Le modèle d'Hammerstein [11] est un modèle populaire en génie biomédical. C'est un modèle de simulation traditionnel, basé sur des formules mathématiques solides et comportant des explications physiques. Le modèle a été utilisé avec succès dans [18, 109] pour la modélisation de la dynamique musculaire isométrique. Le modèle d'Hammerstein contient deux modules en série: un module de non-linéarité précédant un module linéaire de second ordre. Le module non linéaire en entrée représente l'ampleur de la déformation et la dynamique musculaire stimulée est déterminée par le module linéaire. Le modèle d'Hammerstein peut être caractérisé par les paramètres de deux sous-modules:

- $p$  pour le module non linéaire;
- $(\alpha, \beta)$  pour le module linéaire

et par l'erreur d'estimation  $E_H$ .

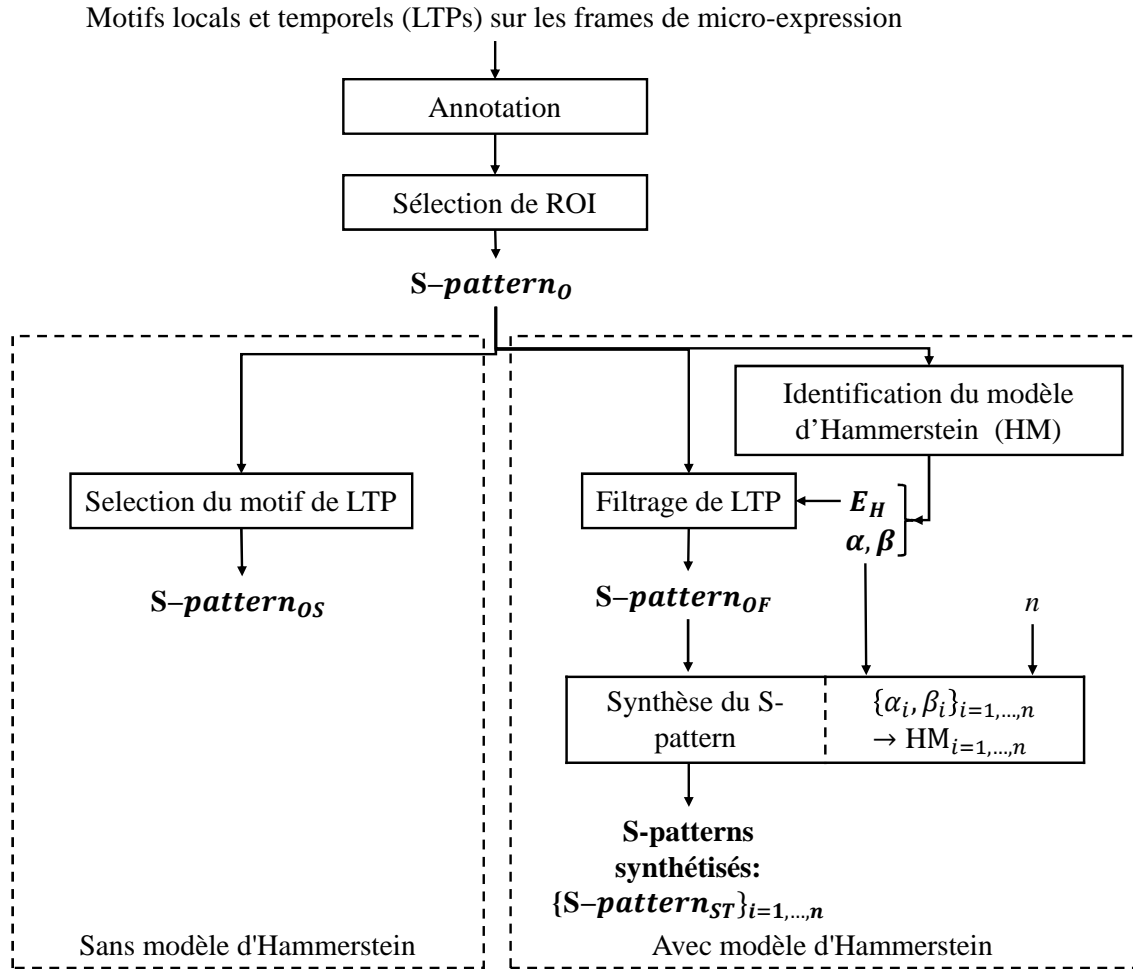


Figure 0-6: Filtrage de LTP et synthèse du S-pattern par le modèle d’Hammerstein (HM) pendant la phase d’entraînement. Dans le bloc de droite, le S-pattern original ( $S\text{-pattern}_O$ ), après l’annotation et la sélection de l’AU de la figure 0-5) passe par l’identification du système du modèle d’Hammerstein. L’ensemble de paramètres  $(\alpha, \beta, E_H)$  correspondant à ce S-pattern est ensuite estimé. Les S-patterns sont sélectionnés par le processus de filtrage de LTP en fonction de l’erreur d’estimation  $E_H$ . Les motifs sélectionnés ( $S\text{-pattern}_{OF}$ ) sont utilisés pour générer  $n$  S-patterns synthétisés ( $S\text{-pattern}_{ST}$ ). Pour la comparaison, le bloc de gauche montre notre méthode sans modèle d’Hammerstein, c’est-à-dire le résultat après la sélection du motif de LTP:  $S\text{-pattern}_{OS}$ .

Les abréviations ci-dessous sont fréquemment utilisées.

$S\text{-pattern}_O$ : S-pattern original après annotation de l’étiquette et sélection de l’AU de la Figure 0-5.

$(\alpha, \beta)$ : paramètres dans le module linéaire du modèle d’Hammerstein.

$E_H$ : Erreur d’estimation du modèle d’Hammerstein

$S\text{-pattern}_{OF}$ : S-pattern original ( $S\text{-pattern}_O$ ) conservé après le filtrage de LTP.

$S\text{-pattern}_{ST}$ : S-pattern synthétisé par le modèle d’Hammerstein.

$S\text{-pattern}_{OS}$ : S-pattern original ( $S\text{-pattern}_O$ ) conservé après la sélection du motif LTP de la Figure 0-5.

## Appliquer le modèle d'Hammerstein au S-pattern

En appliquant le modèle d'Hammerstein à une estimation de S-Pattern, nous nous concentrons sur les deux paramètres de module linéaire et sur l'erreur d'estimation pour synthétiser les S-patterns. En effet, le paramètre  $p$  a peu d'influence sur la propriété dynamique de la micro-expression. Nous nous focalisons, dans les deux dernières sous-sections, sur la distribution de  $(\alpha, \beta)$  du module linéaire et sur l'erreur d'estimation  $E_H$ . Ces paramètres sont associées à la forme de la courbe en S-pattern. L'analyse nous permet de filtrer les S-patterns mal étiquetés par  $E_H$ , puis de synthétiser davantage de S-patterns avec la répartition des  $(\alpha, \beta)$ .

## Filtrage de LTP

Dans cette section, un traitement de filtrage utilisant le modèle d'Hammerstein (filtrage de LTP) est proposé. Ce processus remplace la sélection du motif de LTP à l'étape d'apprentissage de la classification locale. Le processus de filtrage de LTP utilise l'erreur d'estimation  $E_H$  pour filtrer les S-patterns mal étiquetés. Cela permet de conserver les S-patterns les plus fiables pour l'entraînement.

## Synthèse de S-patterns

Ensuite, nous visons à augmenter le nombre de S-patterns pour l'entraînement du classificateur. Les S-patterns avec différentes formes de courbes peuvent être générés par le modèle d'Hammerstein en faisant varier  $(\alpha, \beta)$  autour de paramètre  $(\bar{\alpha}, \bar{\beta})$  appris sur des données réelles.

## Chapitre 5: Résultats expérimentaux

Nous présentons les résultats expérimentaux de notre méthode dans ce chapitre. Tout d'abord, nous présentons la méthode SOA (État-de-l'art) pour la comparaison. Les bases de données, les configurations et les mesures relatives aux expériences sont également présentées.

Pour prouver l'efficacité de notre méthode, nous la comparons avec la méthode SOA: LBP- $\chi^2$ -distance (LBP- $\chi^2$ ) [91]. Notre méthode surpasse la méthode de LBP- $\chi^2$ . Ensuite, les résultats expérimentaux sont analysés concernant nos contributions.

La première contribution principale consiste à détecter les MEs en classant une nouvelle caractéristique: le motif temporel local. Afin de démontrer la pertinence du motif LTP, nous le comparons à une autre caractéristique temporelle utilisée couramment (LBP-TOP). Notre LTP fonctionne mieux que LBP-TOP pour la sélection de MEs. Nous prouvons également la généralité de notre caractéristique proposée: LTP parmi les bases de données différentes. En plus, la performance de détection par émotion et l'analyse statistique de LTP pour les émotions différentes prouvent que le S-pattern est identique pour tous les types d'émotions. Ensuite, nous analysons les paramètres dans deux sous-processus: extraction des ROI dans le pré-traitement et l'ACP dans le calcul de LTP. L'analyse permet de déterminer le réglage optimal du ROI et de prouver l'efficacité de l'ACP.

La deuxième contribution est la fusion spatiale et temporelle. L'étude du processus de fusion montre sa capacité à différencier le ME des autres mouvements. De plus, l'impact des valeurs de seuil dans le processus de fusion est analysé pour trouver les paramètres optimaux.

La troisième est l'autre contribution principale de notre processus: l'augmentation des données à l'aide du modèle d'Hammerstein. La comparaison entre notre méthode avec et sans modèle d'Hammerstein montre que l'augmentation des données améliore les performances de détection. Nous montrons l'efficacité de la synthèse du S-pattern par le modèle d'Hammerstein, en le comparant à la méthode GAN. Nous montrons aussi l'impact du filtrage de LTP et de la synthèse du S-pattern sur l'ensemble du processus. Pour finir, nous trouve la valeur de seuil optimale pour l'erreur d'estimation ( $T_E$ ) et le multiple de génération  $n$  pour la détection de MEs, en étudiant respectivement l'impact de ces paramètres sur deux sous-processus. Enfin, l'analyse de différents modèles de distribution de ( $alpha$ ,  $beta$ ) montre que la quantité de génération de S-pattern importe plus que le choix du modèle de distribution.

La quatrième contribution concerne la détection de la micro-expression dans de longues vidéos utilisant notre méthode. Le résultat de la détection montre que notre méthode sur-



utilise la méthode de base (méthode LBP- $\chi^2$ -distance).

## Conclusion

Nos contributions sont décrites ci-dessous:

- **Une nouvelle caractéristique pertinente pour la détection de la micro-expression: motif temporel local (LTP);**
- Une fusion spatiale et temporelle tardive, qui aide à renforcer la capacité de distinguer les micro-expressions des autres mouvements du visage;
- **Un filtrage de LTP et une augmentation des données par le modèle d'Hammerstein**
- Le premier challenge de détection de micro-expressions: 1. une nouvelle méthode d'évaluation des résultats par intervalle permettant d'évaluer les performances de détection; 2. la détection de micro-expressions dans les longues vidéos par la méthode de base et la méthode proposée

Quatre perspectives sont discutées comme suit:

1). En ce qui concerne la **perspective de notre méthode**, nous discutons des trois points suivants:

a. **Réduction du nombre de faux positifs:** Les recherches à venir devraient se concentrer sur l'amélioration de la capacité de distinguer ME des autres mouvements faciaux.

b. **Schéma de «la détection et la reconnaissance de la micro-expression»:** la position locale des S-patterns détectés peut être utilisée comme caractéristique pour la reconnaissance d'une micro-expression.

c. **Généralisation de la méthode de détection:** Nous soulignons des pistes d'amélioration ultérieure de notre méthode: d'abord, la validation inter-base de données; deuxièmement, la généralisation des paramètres pour différentes bases de données.

2). En ce qui concerne **l'augmentation des données pour la détection de la micro-expression**, étant donné que l'analyse de la micro-expression est limitée par la quan-

tité d'échantillons de micro-expression, l'augmentation des données est nécessaire pour améliorer les performances.

3). En ce qui concerne la **cohérence de la métrique**, comme l'apprentissage automatique est la tendance des recherches pour l'analyse de la micro-expression, le F1-score est recommandé. En outre, détecter des micro-expressions par intervalle semble prometteur car il donne plus d'échantillons pour étudier le mouvement détecté.

4). En ce qui concerne les **applications de détection de micro-expressions**, nous attendons avec intérêt les applications de détection de micro-expressions dans le monde réel, avec un contexte applicatif effectif.



# Contents

<b>Abstract</b>	<b>25</b>
<b>1 Introduction</b>	<b>27</b>
Background and Our Motivation . . . . .	27
Our Contribution . . . . .	29
Thesis Organization . . . . .	31
<b>2 State of Arts</b>	<b>35</b>
2.1 Micro-Expression Databases . . . . .	35
2.1.1 Introduction for Published Micro-Expression Databases . . . . .	36
2.1.2 The 13 Characteristics of Micro-Expression Databases . . . . .	43
2.1.3 Discussion on Micro-Expression Database . . . . .	53
2.1.4 Conclusion for Micro-Expression Databases . . . . .	56
2.2 Result Evaluation Methods and Metrics . . . . .	57
2.2.1 Evaluation of Micro-Expression Recognition Result . . . . .	57
2.2.2 Evaluation of Micro-Expression Spotting Result . . . . .	62
2.2.3 Conclusion for Result Evaluation Methods and Metrics . . . . .	68
2.3 Micro-Expression Spotting Methods . . . . .	70
2.3.1 Related Works Comparison by Algorithms . . . . .	70
2.3.2 Related Works Comparison by Features . . . . .	72
2.3.3 Conclusion . . . . .	73
2.4 Micro-Expression Spot-and-Recognize Schemes . . . . .	74
2.5 Conclusion . . . . .	77

<b>3</b>	<b>Local temporal pattern for Micro-expression Spotting</b>	<b>79</b>
3.1	Pre-Processing . . . . .	81
3.1.1	Facial Landmarks Detection . . . . .	81
3.1.2	Extraction of ROIs . . . . .	81
3.2	Feature extraction: Local Temporal Pattern Computation . . . . .	83
3.2.1	Main Local Movement Extraction by PCA . . . . .	83
3.2.2	Extraction of Local Temporal Pattern . . . . .	87
3.2.3	S-pattern: a Unique Invariant LTP Pattern for All Micro-Expressions	92
3.3	Micro-Expression Spotting: Classification and Fusion . . . . .	95
3.3.1	LTP Selection for Training . . . . .	95
3.3.2	LTP Classification - SVM . . . . .	97
3.3.3	Spatial and Temporal Fusion . . . . .	98
3.4	Spotting Micro-Expression in Long Videos . . . . .	103
3.4.1	Pre-Processing in Long Videos . . . . .	103
3.4.2	Obtaining Spotting Result in a Long Video . . . . .	104
3.5	Conclusion . . . . .	106
<b>4</b>	<b>Data augmentation using Hammerstein Model</b>	<b>107</b>
4.1	Hammerstein Model . . . . .	110
4.2	Applying Hammerstein Model to S-Pattern . . . . .	112
4.2.1	System Identification . . . . .	112
4.2.2	Parameter for the Non-Linear Module . . . . .	114
4.2.3	Relationship between the Linear Module and S-patterns . . . . .	115
4.2.4	Configurations for LTP Filtering and S-pattern Synthesizing . . . . .	117
4.3	LTP Filtering . . . . .	118
4.4	S-pattern Synthesizing . . . . .	121
4.5	Conclusion . . . . .	123
<b>5</b>	<b>Experimental Results</b>	<b>125</b>
5.1	Databases, Comparison Method, Experimental Configuration and Metrics .	127
5.1.1	Databases . . . . .	127

5.1.2	SOA Method For Comparison: LBP- $\chi^2$ -distance . . . . .	128
5.1.3	Parameters Configuration . . . . .	129
5.1.4	Result Evaluation Method and Metrics . . . . .	129
5.2	LTP-ML : An Efficient Method for ME Spotting . . . . .	134
5.2.1	LTP-ML Outperforms SOA LBP- $\chi^2$ Method . . . . .	134
5.2.2	Relevancy of LTP Compared with LBP-TOP for ME Spotting . . .	135
5.2.3	Generalization of LTP for Different Databases . . . . .	137
5.2.4	Unique S-pattern for All Emotions . . . . .	137
5.2.5	Impact of ROI Parameters on Spotting Results . . . . .	140
5.2.6	PCA is More Suitable for Dimension Reduction in Our Method than Auto-Encoder and GPLVM . . . . .	141
5.3	Differentiating ME from Other Facial Movement by Spatial and Temporal Fusion . . . . .	145
5.3.1	Impact of Each Step of the Fusion process on the Spotting Perfor- mance . . . . .	145
5.3.2	Impact of $T_{dist}$ and $T_{CN}$ on the Fusion Process . . . . .	147
5.4	Data Augmentation by Hammerstein Model for ME spotting . . . . .	149
5.4.1	Improvement of Spotting Performance by Data Augmentation Us- ing Hammerstein Model . . . . .	150
5.4.2	Hammerstein Model is More Appropriate than GAN for ME Spotting	150
5.4.3	Analysis Combining LTP Filtering and S-pattern Synthesizing . . .	153
5.4.4	Impact of Threshold Value for LTP Filtering . . . . .	155
5.4.5	Impact of $n$ for S-pattern Synthesizing . . . . .	156
5.4.6	S-pattern Synthesizing by Poisson Distribution of $\alpha$ and $\beta$ . . . . .	157
5.5	Spotting Micro-Expression in Long Videos . . . . .	159
5.6	Conclusion . . . . .	162
<b>6</b>	<b>Conclusion and Perspective</b>	<b>163</b>
6.1	Conclusion . . . . .	163
6.2	Perspectives . . . . .	165

6.2.1	Perspective of Our Method . . . . .	165
6.2.2	Data Augmentation for Micro-Expression Spotting . . . . .	166
6.2.3	Consistency of Metrics . . . . .	166
6.2.4	Micro-Expression Spotting Applications . . . . .	166
<b>Glossary</b>		<b>169</b>
<b>Publication</b>		<b>171</b>
<b>A</b>	<b>FACS - Facial Action Coding System[1]</b>	<b>173</b>
<b>B</b>	<b>Summary Table for Published Micro-Expression Recognition Articles with Their Corresponding Metrics and Databases</b>	<b>177</b>
<b>C</b>	<b>Feature Extraction for Micro-Expression Spotting in Long Videos</b>	<b>179</b>
<b>biblio</b>		<b>181</b>
<b>List of Figures</b>		<b>209</b>
<b>List of Tables</b>		<b>216</b>

# Abstract

## Abstract

The Micro-expressions (MEs) are very important nonverbal communication clues. However, due to their local and short nature, spotting them is challenging. In this thesis, we address this problem by using a dedicated local and temporal pattern (LTP) of facial movement. This pattern has a specific shape (S-pattern) when ME are displayed. Thus, by using a classical classification algorithm (SVM), MEs are distinguished from other facial movements. We also propose a global final fusion analysis on the whole face to improve the distinction between ME (local) and head (global) movements.

However, the learning of S-patterns is limited by the small number of ME databases and the low volume of ME samples. Hammerstein models (HMs) are known to be a good approximation of muscle movements. By approximating each S-pattern with a HM, we can both filter outliers and generate new similar S-patterns. By this way, we perform a data augmentation for S-pattern training dataset and improve the ability to differentiate micro-expressions from other facial movements.

In the first micro-expression spotting challenge of MEGC2019, we took part in the building of the new result evaluation method. In addition, we applied our method to spotting micro-expression in long videos and provided the baseline result for the challenge.

The spotting results, performed on CASMEI and CASMEII, SAMM and CAS(ME)<sup>2</sup>, show that our proposed LTP outperforms the most popular spotting method in terms of F1-score. Adding the fusion process and data augmentation improve even more the spotting performance.



## Résumé

Les micro-expressions (MEs) sont porteuses d'informations non verbales spécifiques, par exemple lors de douleurs. Cependant, de part leur nature locale et brève, il est difficile de les détecter. Dans cette thèse, nous proposons une méthode de détection par reconnaissance d'un motif local et temporel de mouvement du visage. Ce motif a une forme spécifique (motif en S, S-pattern) lorsque la ME apparaît. Ainsi, à l'aide d'un algorithme de classification classique (SVM), nous distinguons les MEs des autres mouvements faciaux. Nous proposons également une analyse de fusion finale globale sur l'ensemble du visage afin d'améliorer la distinction entre les mouvements due à la MEs (locaux) et les mouvements de la tête (globaux).

Cependant, l'apprentissage des S-patterns est limité par le petit nombre de bases de données de ME et par le faible volume d'échantillons de ME. Les modèles de Hammerstein (HM) sont connus pour être une bonne approximation des mouvements musculaires. En approximant chaque S-pattern par un HM, nous pouvons à la fois filtrer les S-patterns réels et générer de nouveaux S-patterns similaires. De cette manière, nous effectuons une augmentation et une fiabilisation des données pour la base de données d'apprentissage de S-patterns et améliorons ainsi la capacité de différencier les MEs d'autres mouvements du visage.

Lors du premier challenge de détection de MEs (MEGC2019), nous avons participé à la création d'une nouvelle méthode d'évaluation des résultats. Cela a aussi été l'occasion d'appliquer notre méthode pour détecter les MEs à longues vidéos. Pour ce challenge, nous avons fourni le résultat de base (baseline) du challenge.

Les expérimentations sont effectuées sur CASMEI, CASMEII, SAMM et CAS(ME)<sup>2</sup>. Les résultats de détection montrent que la méthode proposée surpasse la méthode de détection la plus populaire en termes de F1-score. L'ajout du processus de fusion et de l'augmentation des données améliore encore les performances de détection.

# Chapter 1

## Introduction

### Background

Micro-expression (ME) is a brief local spontaneous facial expression [28], particularly appearing in the case of high psychological pressure. The movement only lasts between  $1/25$  and  $1/5$  of a second [28]. Their involuntary nature helps to affirm they convey the real emotions of a person [26]. This kind of facial expression is a very important non-verbal communication clue. It can reveal the genuine emotion and the personal psychological states [12]. Thus, ME detection and recognition (MEDR) has many potential applications in national security [26], medical care [30], educational psychology [17], and political psychology [108]. For example, by analyzing ME, doctors may observe the level of pain [30], psychologists could find indications of suicide [28].

MEs were discovered by Haggard and Isaacs in 1966 [39] and then, named by Ekman and Friesen [28] in 1969. Ekman developed a first ME training tool: Micro Expressions Training Tools (METT) in 2002 [25]. The tool has several visual samples which belong to the universal emotions. This tool aims at training human beings to detect and interpret MEs. Yet, the overall recognition rate for the 6 basic emotions by human naked eyes is lower than 50%, even by a trained expert [31].

Since the 2000s, research on automatic spotting and recognition of ME (MESR) has developed. Figure 1-1 shows the trend of the number of the MESR research articles. This number is low but is increasing, and we can notice there are much more ME recognition

papers than ME spotting ones. Thus, the recognition rate is getting higher, e.g. 86.35% of accuracy for 5 classes in [20]. Yet, most of ME recognition methods are performed between the onset and offset frame. But finding onset and offset frames, especially when MEs are spontaneous micro facial movements in a whole video sequence, is still a huge challenge.

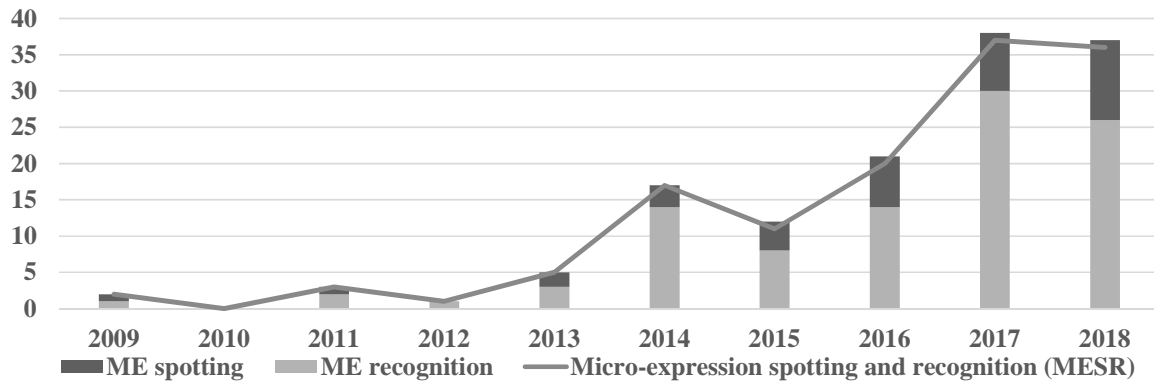


Figure 1-1: MESR research trend. The number of articles on MESR is increasing by year, mainly in the area of ME recognition (bottom column). ME spotting research has not yet attracted sufficient attention (column at the top).

The spotting results of current proposed methods are not accurate enough, certainly because of the ME nature and also of the limited number of public ME databases. For instance, in the spotting task of the Second Facial Micro-Expression Grand Challenge (MEGC2019) [64], the baseline of f1-score is less than 0.05. Even though the ME samples are produced in a strictly controlled environment, there are many false positives due to head movement or eye blinking. It is challenging to differentiate ME from them.

Micro-expression analysis is limited by the small feature volume. It is due to the size of ME databases. For instance, the largest spontaneous micro-expression database only contains 255 video samples. This situation largely restricts the utilization of machine learning for ME spotting.

Another unsolved problem for micro-expression spotting is that the metrics used to analyze the result in different papers are divers. The spotting results are studied per frame, per interval or per video, while the metric could be TPR, ROC, ACC, F1-score and other measures. Indeed, researchers try to provide as many metrics as possible in the article to analyze comprehensively the method. The used metrics are chosen based on the proposed method. As the methods are not the same, each paper uses different metrics from the other

papers. It is difficult to define one metric rather than another.

## Our Contributions

We explore an automatic system for spotting MEs which could:

- spot micro-expression frames in video sequences
- separate motions related to MEs from head movement or eye blinking.
- detect the region where the ME occurs.
- increase the size of ME feature dataset for training step in machine learning.

We have four contributions in this article. The main one is to propose a dedicated **local temporal pattern (LTP)**, which extracts relevant information for ME spotting [63]. Since ME is a brief and local movement, the motion pattern is analyzed from local face regions. The locations are the small regions of interest (ROIs) where MEs can occur. When there is an ME, the texture (i.e. the grey level value) of ROI changes. Hence, we calculate this change, which is called the distance between two grey level ROI frames in our document. One originality of the approach is the utilization of a temporal pattern on those ROIs. The duration is the one of ME (300ms). The curve in Figure 1-2 represents the local temporal variation on grey level texture when a ME is occurring in the ROI. We can notice it forms an S-pattern from the onset to the apex of the ME. This S-pattern appears each time a ME occurs in a region and is independent of the ROI and of the subject, which makes it relevant for ME spotting. More precisely, this LTP pattern is computed over an interval of the duration of a ME (300ms) and the pattern is a list of distances between the grey level textures of the first frame and of the  $k_{th}$  frame of the interval. In order to conserve the most significant variation, principal component analysis (PCA) is first performed (for the corresponding ROI) on the whole video.

The second contribution concerns the elimination of motions related to head movement or eye blinking. The particularity of the approach is the combination of local and global treatments. The LTP spotting is done locally by ROI. A fusion system on the entire face

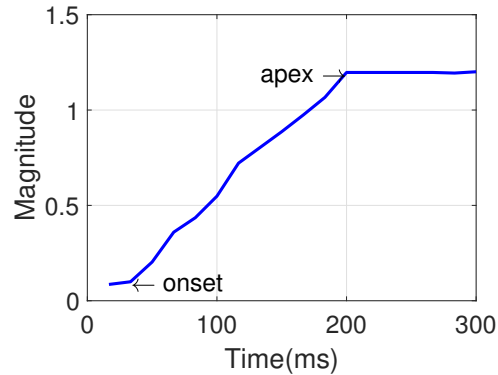


Figure 1-2: Example of Local temporal pattern (LTP) during a ME located in the right eyebrow region over a period of 300ms (average duration of ME). This LTP represents the evolution of the grey level texture of one ROI during the ME. The curve reaches the top in around 150ms and then stays stable or slightly declines. This pattern is specific of ME movements, and is referred to S-pattern due to the curve shape. (Video: Sub01\_EP01\_5 of CASME I)

then separates MEs from other facial movements. Finally, depending on the spotted S-patterns, the time index of the onset of the ME can be determined. Note that as the LTP is local, we also have the information of the location where ME occurs.

Another main contribution concerns spotting improvement by both **filtering wrong S-patterns and data augmentation**. Indeed, the sample amounts in existing ME databases are small, which largely limits the improvement of ME spotting performance. Moreover, databases are not labelled with the precise location of ME (left or right part of the face is not mentioned for example). This leads to wrong annotations of LTP-patterns. The originality is to use Hammerstein model (HM), which is known to approximate efficiently muscle movements. Each S-pattern can be approximated by a HM with two parameters. Using the distribution of those two parameters, we can filter wrong patterns and generate other similar S-patterns. The utilization of those models contributes to the extension and the reliability of the data samples for S-pattern training. Training is then performed on both real and synthesized patterns.

The last contribution is the spotting task of the second Micro-Expression Grand Challenge (MEGC2019). We provided the baseline method and result by spotting ME on long videos sequences. In addition, for the consistency of result evaluation method, we provide a set of new performance metrics. It has been used as the guideline for result evaluation on

## Thesis Organization

The document is organized as follows:

- **Chapter 2** conducts a survey on the state of arts of automatic facial micro-expression spotting and recognition (MESR). First of all, we introduce a systematic analysis of micro-expression databases. Secondly, a review on result evaluation methods and metrics for micro-expression analysis is presented. Thirdly, all published spotting methods are analyzed with their merits and demerits. Then, the micro-expression spot-and-recognize schema is discussed. Finally, we gives our perspective on this research domain and the contributions of our method.
- **Chapter 3** propose our method to spot micro-expression. It focus on our novel relevant feature for micro-expression spotting: local temporal pattern (LTP). Another main point is the late spatial-and-temporal fusion, which is performed to obtain the final micro-expression spotting result. The proposed method consists of three parts: a pre-processing to precisely detect facial landmarks and extract the regions of interest (ROIs), then the computation of local temporal pattern (LTP) on these ROIs and eventually the spotting of micro-expressions. The first three sections in this chapter present these sub-steps of our method applied in short videos. Our method is then adapted to the situations of long videos. Finally, we conclude the chapter and points out the requirements to improve the performance of our method.
- **Chapter 4** proposes to increase the size of S-pattern dataset for the training stage of the local classification, because the small volume of S-pattern dataset limits the performance of our method. To that purpose, we use Hammerstein model (HM). After a brief introduction of Hammerstein model, the model is applied to S-pattern. We find that the parameters  $(\alpha, \beta)$  of the linear module and the estimation error  $E_H$  are associate with the dynamic property of micro-expressions (curve shape of S-pattern). The LTP filtering can use  $E_H$  to filter unreliable S-patterns. Meanwhile, more S-patterns

can be synthesized based on the  $(\alpha, \beta)$  distribution. A brief conclusion is given at the end of this chapter.

- **Chapter 5** presents the experimental results of our method. Firstly, we introduce the state-of-arts (SOA) method for comparison. The databases, configurations and metrics for the experiments are also presented.

To prove the efficiency of our method, we compare it with the SOA method: LBP- $\chi^2$ -distance method [91]. Then, the experimental results are analyzed regarding our contributions.

The first main contribution is to spot ME by classifying a novel relevant feature: local temporal pattern. In order to demonstrate the relevancy of the LTP pattern, we compares it with another common used temporal feature (LBP-TOP). We also proves the generality of our proposed LTP feature among different databases. As we mentions that S-pattern is identical for all kinds of emotions, the spotting performance per emotion and the statistic analysis of LTP for different emotions are investigated to prove the theory Then, we analyse parameters in two sub-processes: extraction of ROIs in pre-processing and PCA in LTP computation. The analysis allows to find the optimal ROI setting and to prove the effectiveness of PCA.

The second contribution is the spatial and temporal fusion. The fusion process is investigated to show its capacity to differentiate ME from other movements. As well, the impact of threshold values in the fusion process are analyzed to find the optimal parameters.

The third is the other main contribution of our process: data augmentation using Hammerstein model. Experiments are performed with and without Hammerstein model, and the results show that the data augmentation improves the spotting performance. To show the effectiveness of S-pattern synthesizing by Hammerstein model, the method is compared with GAN. The analysis shows the impact of LTP filtering and S-pattern synthesizing on the entire process. Meanwhile, in order to find the optimal threshold value for estimation error ( $T_E$ ) and generation multiple  $n$  for ME spotting, the impact of these parameters on two sub-processes are investigated

respectively. Finally, the analyze of different distribution models of  $(\alpha, \beta)$  shows that the generation amount of S-pattern matters more than the choice of distribution model.

The fourth contribution concerns spotting micro-expression in long videos utilizing our method. The spotting result show that our method outperforms the baseline method (LBP- $\chi^2$ -distance method).

- **Chapter 6** summarizes our contributions in this thesis and presents the perspectives of future work.





# Chapter 2

## State of Arts

As presented in Chapter 1, facial micro-expression (ME) analysis has emerged in last 10 years. In this chapter, a survey on the state of arts of automatic facial micro-expression spotting and recognition (MESR) is conducted. First of all, a systematic analysis of micro-expression databases is given in Section 2.1. A review on result evaluation methods and metrics for micro-expression analysis is then presented in Section 2.2. In section 2.3, all published spotting methods are analyzed with their merits and demerits. Then, section 2.4 discusses micro-expression spot-and-recognize schema. Finally, Section 2.5 gives our perspective on this research domain and the contributions of our method.

### 2.1 Micro-Expression Databases

As shown in Table 2.1, unlike large amounts of macro-expression databases, there are only 15 published micro-expression databases. Besides the limited amount, the samples were recorded under various condition for each database. In this section, we propose an all-inclusive survey and comparison on for these published micro-expression databases. Subsection 2.1.1 shows the detailed information for all published micro-expression databases. Then, subsection 2.1.2 presents a systematic analysis based on 13 characteristics. In addition, a discussion is given in subsection 2.1.3 for the further micro-expression database creation. Finally subsection 2.1.4 concludes this section.

Table 2.1: Database amount for macro- and micro-expression.

	Posed	Spontaneous	In-the-wild	Total
Macro-expression [127]	26	27	13	61 <sup>1</sup>
Micro-expression	3	10	2	15 <sup>2</sup>

<sup>1</sup> Some macro-expression databases contain both posed and spontaneous samples.

<sup>2</sup> Two micro-expression databases are mentioned in two published articles, but the databases are not public yet.

### 2.1.1 Introduction for Published Micro-Expression Databases

In this subsection, the development history and basic information of the databases are presented. In Table 2.2, the micro-expression databases are classified into three categories: posed, spontaneous and in-the-wild databases. Then they are sorted depending on the publish year. For a purpose of clarity, databases listed in this thesis are given by their abbreviation.

Table 2.2: Reference of published micro-expression databases.

Expression type	Database	Reference
Posed	Polikovsky’s Database	[101]
	USF-HD	[106]
	MoblieDB	[44]
Spontaneous	York-DDT	[126]
	SMIC-sub	[100]
	CASME I	[137]
	SMIC	[66]
	CASME II	[135]
	Silesian Deception Database	[105]
	SMIC-E	[67]
	CAS(ME) <sup>2</sup>	[104]
	Grobova’s database	[34]
	SAMM	[19]
In-The-Wild	Canal9	[112]
	MEVIEW	[52]

Since 2009, four databases: Canal9 [112], York-DDT [126], Polikovsky’s database [101] and USF-HD [106] were published. However, these databases are not used nowadays. Canal9 and York-DDT do not dedicate to the research of automatic micro-expression analysis: one is for analysis of social interaction, and the other is created for psychological

study for a deception detection test. Meanwhile, the Polikvsky's database and USF-HD are posed ME databases. In the ensuing years, several spontaneous ME databases were created. In 2011 and 2013, the research group of Oulu University published SMIC-sub [100], SMIC [66]. During the same period, CASME I (2013) [137], CASME II (2014) [135] were created by Chinese Academy of Science. In 2015, Radlak et al. built a Silesian Deception Database [105], which provided video samples of deceivers and truth-tellers. In 2017, four public databases are published, including three spontaneous databases and one In-the-wild database. Oulu University published an extended version of SMIC: SMIC-E [67] to provide video samples for ME spotting. Afterwards, Davison et al. created SAMM [19], which is a spontaneous micro-facial movement dataset. Meanwhile, a database which contains both macro and micro expression: CAS(ME)<sup>2</sup> [104] was published. Furthermore, Huasak et al. published an in-the-wild database MEVIEW [52]. In addition, there are two private databases: MobileDB and Grobova's database, which were mentioned in [44] and [34] respectively but are not yet publicly available.

Table 2.3, 2.4 and 2.5 list a comprehensive summary of all the databases. Databases are sorted by alphabetical order. These three tables contain the essential characteristics of a micro-expression database. Further comparison are presented in the following subsections.

Table 2.3: Characteristics summary for micro-expression databases - part 1. Databases are sorted by alphabetical order. The following formatting distinguishes databases: normal for posed databases, bold for spontaneous database, italic for in-the-wild databases, † means that the database is not available online. MaE: macro-expression, PSR: participants' self-report.

	<i>Canal9</i>	<b>CASME1</b>		<b>CASME2</b>	<b>CAS(ME)<sup>2</sup></b>	
		Section A	Section B		partA	partB
Publish year	2009	2013		2014	2017	
Eliciting method	In-the-wild	Neutralization paradigm				
Gender (Female/male)	25/165	13/22-4/15		15/11	13/9	
Age range	N/A	22.03 (SD = 1.60)		22.03 (SD = 1.60)	22.59(SD=2.2)	
Ethnic group(s)	N/A	1		1	1	
# of subjects	190	35-19		35-26	22	
# of samples	70	96	101	255	87	300 MaE 57 ME
# of camera	1	1		1	1	
Background	In-the-wild	Room, white background				
Lightning	N/A	Natural light	2 LED lights	4 LED* <sup>1</sup>	2 LED lights	
FPS	25	60		200	30	
Resolution	720*576	1280*720	640*480	640*480	640*480	
Facial resolution (approximate)	N/A	150*190		280*340	200*240	
Labeling-method	N/A	Two coders and PSR after experiments		Two coders	Two coders and PSR after experiments	
Action Units	No	Yes		Yes	Yes	
Emotional labels	N/A	7	8	5	4	
Average video duration	N/A	2,82s		1.3s	148s	onset-offset

Table 2.4: Characteristics summary for micro-expression databases - part 2. Databases are sorted by alphabetical order. The following formatting distinguishes databases: normal for posed databases, bold for spontaneous database, italic for in-the-wild databases, † means that the database is not available online. Neutralization paradigm: NP, PD:Polikovskiy’s Database, GD: Grobova’s database, PSR: participants’ self-report.

	<b>GD</b> †	<i>MEVIEW</i>	MobileDB†	PD	<b>SAMM</b>	<b>SDD</b>
Publish year	2017	2017	2017	2009	2016	2015
Eliciting method	Simulated expressions	NP	Lie generation	NP or mask	In-the-wild	Simulated expressions
Gender (Female/male)	8/5	2/14	14/3	N/A	16/16	N/A
Age range	20-22	N/A	N/A	25(SD=4)	33.24 (SD = 11.32)	Faculty students
Ethic groups	N/A	1	N/A	3	13	N/A
# of subjects	13	16	17	10	32	101
# of samples	13	31	306	42	224	182
# of camera	1	1	1 ipad air	1	1	1
Background	Room, white background	In-the-wild	Indoor	Uniform	white wall	N/A
Lightning	N/A	N/A	N/A	3 lights*5	2 lights*6	N/A
FPS	100	25	120	200	200	100
Resolution	640*480	720*1280	150*150	480*640	2040*1088	640*480
Facial resolution (approximate)	N/A		150*150	N/A	400*400	N/A
Labeling-method	N/A	One annotator	Mimiced expression	Mimiced expression	Two coders and PSR before experiments	1 video by 1 coder
Action Units	no	yes	no	Yes	Yes	No
Emotional labels	3*2	6	6	6	7/8*7	1*8
Average video duration	1.6-2s	3s	30 frames*3	0.51s (SD=0.2s)	35,3s	N/A

Table 2.5: Characteristics summary for micro-expression databases - part 3. The following formatting distinguishes databases: normal for posed databases, bold for spontaneous database, italic for in-the-wild databases. PSR: participants' self-report. HS: high speed camera, VIS: a normal visual camera, NIR: a near-infrared,

	SMIC-sub	SMIC/SMIC-E			USF-HD	York-DDT
		HS	VIS	NIR		
Publish year	2011	2013/2015			2011	2009
Eliciting method	Neutralization paradigm					
Gender (Female/male)	3/3	6/14				31/19=1.63
Age range	N/A	26.7(22-34)			N/A	18-45
Ethnic groups	N/A	3			N/A	N/A
# of subjects	6	16	8			9(6/3)
# of samples	77	164/157	71			181 MaE, 100 ME
# of camera	1	3*4			1	1
Background	An indoor bunker environment					
Lightning	N/A	4 lights from the four upper corners of the room				N/A
FPS	100	100	25			25
Resolution	640*480	640*480			720*1280	320*240
Facial resolution (approximate)	190*230					
Labeling-method	Two coders according to PSR emotions	Two coders and PSR after experiments				Encoded by 20 participants
Action Units	No	No				No
Emotional labels	2/5*9	3			6	5
Average video duration	onset-offset	1st frame: neutral face, 2nd frame: ME/ 5.9s				60 s

- \*<sup>1</sup>: Four LED under umbrella reflectors.
- \*<sup>2</sup>: Sad (obvious sadness), neutral and blocked (hidden sadness).
- \*<sup>3</sup>: The videos are normalized to 30 frames.
- \*<sup>4</sup>: A high speed (HS) camera, a normal visual camera (VIS) and a near-infrared (NIR).
- \*<sup>5</sup>: Three lights were used for shadow cancellation, and diffusion sheets were used to minimize hot spots on the facial image.
- \*<sup>6</sup>: Two lights that contain an array of LEDs was used. DC (direct current) source was used to avoid flickering. Light diffusers were placed around the lights to soften and even out the light on participants' faces.
- \*<sup>7</sup>: Video samples in SAMM database were divided into 7 objective classes. And the number of emotional classes is 8.
- \*<sup>8</sup>: The database contains eye closures, gaze aversion including saccadic eye movements and facial distortions.
- \*<sup>9</sup>: The database contains two types of emotional labels. One is positive and negative. The other one is happy, sad, disgust, surprise and fear.

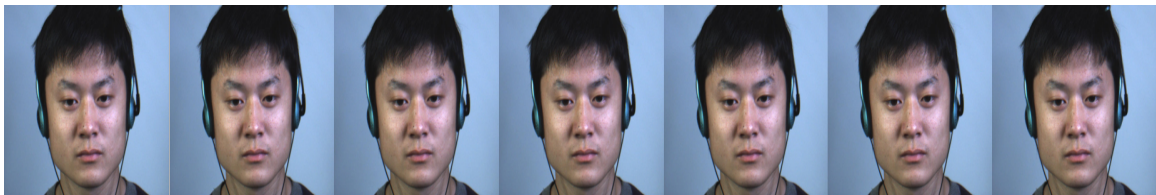
### **Samples in Spontaneous Micro-Expression Database**

The samples in spontaneous micro-expression databases are shown in Figure 2-1. CASME II and SAMM provide raw images from video sequences. Meanwhile, CASME I, SMIC and CAS(ME)<sup>2</sup>, also contain cropped images, i.e. the facial region image.





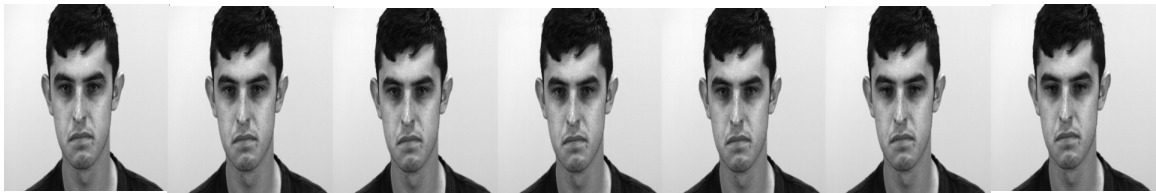
(a) CASME I



(b) CASME II



(c) SMIC



(d) SAMM



(e) CAS(ME)<sup>2</sup>

Figure 2-1: Samples in spontaneous micro-expression database

## 2.1.2 The 13 Characteristics of Micro-Expression Databases

Instead of listing the information for each database respectively, comparison depending on certain characteristics can indicate more clearly their merits and demerits. A classification system based on chosen characteristics facilitates the database selection for micro-expression analysis. In addition, it points out the direction for future micro-expression database creation.

Thus, in this sub-section, we list 13 characteristics which can comprehensively represent the feature of ME databases. They are classified into four categories, as set out in Table 2.6, including population, hardware, experimental protocol and annotations. This classification is inspired by Weber et al. [128], and has been adapted to be more appropriate for ME databases description. These four categories will be presented comprehensively in following sub-subsections.

Table 2.6: Categories and characteristics of ME databases. The characteristics are coded to simplify further representation.

Category	Characteristic	Code
Population	# of subjects	P.1
	# of samples	P.2
	Gender (%)	P.3
	Age range	P.4
	Ethnic group(s)	P.5
Hardware	Modalities	H.1
	FPS	H.2
	Resolution	H.3
Experimental protocol	Method of acquisition	EP.1
	Environment (Image/video quality)	EP.2
	Available expressions	EP.3
Annotations	Action Units	A.1
	Emotional labels	A.2

### Population

The first category: population contains the basic characteristics for databases, which concern the participants' information. This analysis of population focuses on the number of

subjects (P.1), the number of samples (P.2), the gender distribution (P.3), the age range (P.4) and the ethnic groups (P.5). A ME has a general variation pattern, but also differs for different subjects, because of the shape of their face, their texture, their gender and the cultural influence. For instance, eastern people express their emotions with dynamic eye activity, while western people use more mouth area [53]. As a result, the genericity of ME database, i.e. the amount of subjects, a large age range, uniform women/men distribution and wide ethnic groups, is essential for improving the ability of automatic ME spotting and recognition.

As shown in Table 2.7, most of ME databases contain less than 50 subjects (P.1), whether they are posed, spontaneous or in-the-wild. Moreover, the amount of ME samples (P.2) is not significant. Even the largest database CASME II [135] does not exceed 255 samples, which make it difficult to train spotting or recognition algorithms. This is because the ME samples are difficult to produce. It requires a strict recording environment and professional eliciting methods. Moreover, the annotation is time-consuming. Besides, even though the ME exists in our daily life, it is complicated to gather video samples and to identify the facial movement precisely in the in-the-wild environment.

The women/man percentage (P.3) for ME databases is not well balanced. Canal9 [112], CASME I [137], SMIC [66] and MEVIEW [52] contain much more male subjects than female, while the number of female subjects in York-DDT [126] is almost two times the male subject amount. Yet, the percentage in the three most recent databases CASME II [135], SAMM [19] and CAS(ME)<sup>2</sup> [104] are well balanced between 40/60 and 60/40. (See Table 2.3 and Table 2.4 in Appendix for exact values)

The age range (P.4) for most ME databases is quite low, since the majority of samples were produced by volunteers in university. The average age is around 25 years old and the standard deviation (STD) is around 3. Yet, York-DDT [126] has a moderate range (18-45), and the average age of SAMM [19] is 33.24 with a large STD (11.32). However, the age distribution is still far from the reality. A good database should also contain the samples gathered from children and elderly people.

For ME database, the ethnic groups (P.5) are not very diverse. China Academy of Science (CAS) has built three databases: CASME I [137], CASME II [135] and CAS(ME)<sup>2</sup> [104],

Table 2.7: Classification of the databases according to the characteristic P.1, P.2, H.1, A.1 and A.2 (# of subjects, # of samples, modalities, action units and emotional labels). Databases are sorted by alphabetical order. The following formatting distinguishes databases: normal for posed databases, bold for spontaneous database, italic for in-the-wild databases, \* means the database is not available online. 2D V: 2D video. SMIC and SMIC-E both have three sub-classes: NIR, VIS and HS. Sub-class HS of SMIC / SMIC-E is separated from the other two because of the different number of ME video samples.

Databases	P.1	P.2	H.1	A.1	A.2
<i>Canal9</i>	$\in (200, 250)$	$\in (50, 100)$	2D V		
<b>CASME I</b>	$\leq 50$	$\in (100, 200)$	2D V	✓	✓
<b>CASME II</b>	$\leq 50$	$\in (200, 300)$	2D V	✓	✓
<b>CAS(ME)<sup>2</sup></b>	$\leq 50$	$\in (50, 100)$	2D V	✓	✓
<b>Grobova's database*</b>	$\leq 50$	$\in (50, 100)$	2D V		✓
<i>MEVIEW</i>	$\leq 50$	$\leq 50$	2D V	✓	✓
MobileDB*	$\leq 50$	$\in (200, 300)$	2D V		✓
Polikovsky's Database	$\leq 50$	$\leq 50$	2D V	✓	
<b>SAMM</b>	$\leq 50$	$\in (100, 200)$	2D V	✓	✓
<b>Silesian Deception</b>	$\in (100, 200)$	$\in (100, 200)$	2D V		✓
<b>SMIC-sub</b>	$\leq 50$	$\in (50, 100)$	2D V		✓
<b>SMIC-NIR, VIS</b>	$\leq 50$	$\in (50, 100)$	2D V + IF		✓
<b>SMIC-HS</b>	$\leq 50$	$\in (100, 200)$	2D V		✓
<b>SMIC-E-NIR, VIS</b>	$\leq 50$	$\in (50, 100)$	2D V + IF		✓
<b>SMIC-E-HS</b>	$\leq 50$	$\in (100, 200)$	2D V		✓
USF-HD	$\leq 50$	$\in (100, 200)$	2D V		✓
<b>York-DDT</b>	$\in (50, 100)$	$\leq 50$	2D V		✓

but there is only one ethnic group: Asian. Meanwhile, SMIC [66] has 3 ethnic groups: Caucasian, Asian and Africa, and Polikovsky’s database [101] have Caucasian, Asian and Indian groups. Furthermore, SAMM [19] contains 13 ethnic groups, which makes it the most varied ME database in term of ethnic groups (P.5). A widely collected database is recommended for ME analysis in the real world. Yet, the construction of this kind of database may need the international cooperation.

## Hardware

In this part, this category: hardware links to the necessary information of video samples. The characteristics related to hardware, i.e. modalities, resolution and FPS, are discussed. The first characteristic is modalities (H.1), which means the ME sample recorded format. Until now, as listed in Table 2.7, the modality for most ME databases is unified: a unimodal 2D video. However, SMIC [66] and SMIC-E [67] have multi-modalities, with three different 2D videos: high speed (HS) video, normal visual (VIS) video and near-infrared (NIR) video, as shown in Figure 2-2. Multi-modalities (e.g. facial thermal variation from infrared images) can allow the analysis methods to extract more different features and therefore enhance the reliability of emotion classification. Meanwhile, the synchronization should catch our attention. Compared with macro-expression databases [128], there is no audio, 3D model, or body movements in the databases. If the ME databases follow the same evolution as the macro-expression databases, we can imagine having more modalities in the future databases.

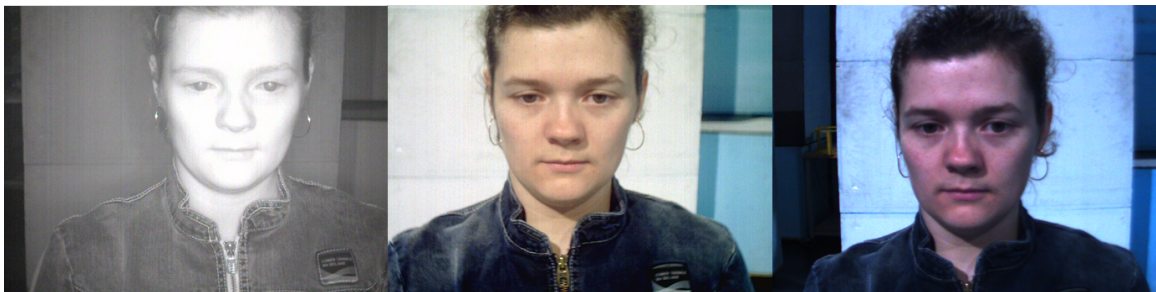


Figure 2-2: Three Modalities of SMIC database. Sample at the left side is NIR image, in the middle is the VIS image and at the right side is the HS image.

As the ME average duration is around 300ms [63] and the movement usually appears on

the local facial regions, a high FPS (H.2) and a high resolution (H.3) will help to capture the subtle facial movements. Table 2.8 lists the frame amounts for a micro-expression with an average duration (300ms) depending on different FPS. Most ME databases have at least 60 FPS with a facial resolution larger than  $150 \times 190$ . The FPS of Polikovsky’s Database [101], CASME II [135] and SAMM [19] reach to 200. The resolution of the facial region in SAMM [19] is  $400 \times 400$ . These ME sequences were recorded by a high-speed camera in a strictly controlled laboratory environment to reduce the noise. Meanwhile, USF-HD [106], SMIC [66], CAS(ME)<sup>2</sup> [104] and MEVIEW [52] contain clips with low FPS, lower or equal to 30. These databases fit more the situation in real life. However, depending on the average duration of ME, 30 fps means that the video just contains 9 frames for ME (300ms). Thus, the data scale is small, and this may make the ME analysis more complex and less reliable.

Table 2.8: Number of frames for a micro-expression with average duration (300ms) depending on different FPS.

FPS	30	60	100	200
NbFr <sub>ME</sub>	9	18	30	60

## Experimental Protocol

The experimental protocol refers to the acquisition method (EP.1), experimental environment (image/video quality) (EP.2) and the available expressions (EP.3). As the protocols are quite different according to the type of database: posed, spontaneous and in-the-wild, we discuss them separately in the following paragraphs. Moreover, as image/video quality is a very important factor for ME databases, this characteristic is specifically discussed in Paragraph *Image/Video Quality*.

**Posed Micro-Expressions** Posed ME means that the facial movement is expressed by a subject on purpose with simulated emotion. If we look at the macro-expression, there are three methods of reproduction: free reproduction, ordered reproduction and portrayal [128] which means the subject is required to improvise on an emotionally rich scenario. But things are different for micro-expressions. ME is challenging to produce because ME is a

very brief and local facial movement. The ME sequences in Polikovsky's Database [101], USF-HD [106] and mobileDB [44] are all reproduced by ordered reproduction (EP.1). In the Polikovsky's Database, volunteers were requested to perform 7 basic emotions slightly and quickly after being trained by an expert. Subjects in USF-HD were demanded to mimic the MEs in sample video and the participants in mobileDB mimicked the expressions based on 6 MEs (i.e. happiness, surprise, fear, sadness, disgust and anger).

The experimental environment (image/video quality) (EP.2) contains the number of cameras, background, lighting conditions and occlusions. For existing ME databases, there is only one camera facing the subject. As a side note, the video samples in mobileDB were recorded by a mobile device, which could be used for daily emergency situations. Regarding background and lighting condition, it is the same situation for posed and spontaneous databases: an indoor environment with uniform lighting. Concerning occlusions, almost all the databases contain subjects wearing glasses. However, other occlusions and the head pose variation are very rare in ME databases.

Joy, sadness, surprise, fear, anger and disgust are six basic emotions [29]. They are regarded as available expression (EP.3) in these three databases. The Polikovsky's Database has one more emotional content: contempt. Moreover, the videos in this database are FACS-coded.

***Spontaneous Micro-Expressions*** Spontaneous ME is generated naturally by emotion affect. All the spontaneous ME database used passive task as the emotion elicitation method (EP.1). The most common method is the neutralization paradigm, i.e. asking participants to watch videos containing strong emotions and try to neutralize during the whole time or try to suppress facial expressions when they realized there is one. The samples in York-DDT [126] and Silesian Deception database [105] were generated by lie generation activity. Moreover, there is another kind of micro-expression, which is masked ME. It is hidden behind other facial movements, and it is more complicated than neutralized ME. We will discuss the masked ME in section 2.1.3.

Regarding the experimental environment (EP.2), it is quite similar to that of pose MEs, except that SMIC [66] has three cameras: a high speed (HS) camera, a normal visual

camera (VIS) and a near-infrared (NIR) camera. Most of the databases have only one lightning condition. CASME I [137] is the exception. The database is divided into two sections for two different lighting conditions: natural light and two LED lights.

As already introduced for characteristic H.1, ME database modality is 2D video. The duration of video sequences is quite short: most videos are less than 10s (see Table 2.3 and Table 2.4 for more details). For ME recognition, most methods only use the frames between the onset and the offset. Yet, longer video, with sometimes several MEs, is better for ME spotting. SMIC-E [67] provided a longer version of video samples in SMIC [66], SAMM [19] contains videos with a long duration, the average time is 35.3s. In CAS(ME)<sup>2</sup> [104], the longest video can reach to 148s.

Concerning available expressions (EP.3), there are two classification methods. One is respecting the 6 basic emotion classes, e.g. York-DDT [126], CASME I [137], CASME II [135] and SAMM [19]. The other one is classifying emotions into three or four classes: positive, negative, surprise and others, such as SMIC [66], SMIC-E [67] and CAS(ME)<sup>2</sup> [104]. In addition, Silesian Deception Database [105], SAMM [19] and CAS(ME)<sup>2</sup> [104] consist of not only micro movements but also macro expressions.

***In-The-Wild Micro-Expressions*** In-the-wild ME means that the acquisition is not limited by population and experiment acquisition conditions (EP.1). There are only 2 in-the-wild ME databases: Canal9 [112] and MEVIEW [52]. They both consist of a corpus of videos of spontaneous expressions. Canal9 [112] contains 70 political debates recorded by the Canal9 local station. ME can be found when the politicians try to conceal their real emotion. MEVIEW [52] contains 31 video clips from poker games and TV interviews downloaded from the Internet. The poker game can help to trigger ME thanks to the stress and the need to hide emotions. Image samples are shown in Figure 2-3. For the experimental environment (EP.2), it is worth noting that the facial area in MEVIEW [52] varies because the camera is often zooming, as well as changing the angle and the scene. For instance, the upper images in Figure 2-3 illustrate the subject with different facial resolutions because the camera zoomed in. Furthermore, as most videos came from television programs, there are some body movements and head poses. The available expressions (EP.3) in these two



databases are based on 6 basic emotions. It is a big challenge to detect and recognize the ME automatically since there are a lot of other irrelevant movements.



Figure 2-3: Images samples from MEVIEW database

**Image/Video Quality** Image/video quality is a very important aspect of facial expression analysis. This subsection is dedicated to the discussion of this subject. It already exists various macro-expression databases which contain different image quality situations. Unfortunately, the majority of published ME databases are spontaneous ME databases. Video samples are recorded in a strictly controlled laboratory environment. Figure 2-4 [135] shows a typical acquisition setup for recording micro-expression samples. The improvement of the latest published databases focuses more on population augmentation and video length rather than image quality. The illumination condition is maintained to be stable. LED lights are commonly used, and in some cases, researchers used extra equipment to reduce the noise. For instance, in SAMM database [19], light diffusers were placed around the lights to soften the light on the participants' faces. The background is normally white or gray. Besides, to avoid unrelated movements, participants were required to stay still and face directly the camera. The most recent published database MEVIEW [52] is an in-the-wild database. The video samples were gathered from poker-game television shows.

The image quality varies when the camera zooms in or out. Moreover, it is still challenging to accurately spot and recognize ME in single viewing angle videos with little noises. Thus, the community has not paid sufficient attention to get various image quality situations. However, as it is an essential factor for ordinary facial expression databases, we could expect its importance in future ME databases.



Figure 2-4: Acquisition setup for elicitation and recording of micro-expressions [135]

## Annotations

This characteristics in this category directly link to automatic micro-expression analysis. The evaluation of prediction result is based on annotations (ground truth) of databases. Regarding of ME databases, low-level information: action units (A.1) and high-level information: emotional labels (A.2) are the two major annotations.

Facial Action Coding System (FACS) [27] is an essential tool for facial expression annotation. Indeed, the facial components of FACS, i.e. actions units (AUs), identify the local muscle movement, and the combination of AUs shows the emotional expression, e.g. AU6+AU12 (Cheek raiser and Lip corner puller) could indicate happiness. Since ME is a local brief movement, identifying the AUs will help to facilitate the spotting and recognition for ME. However, some databases were not labeled by AUs, e.g., USF-HD [106], SMIC [66] and SMIC-E [67].

Figure 2-5 shows a histogram for the sum of AU annotations in all the databases and lists the number of AUs annotation in ME databases. The highest AU amount represents the regions where have the most ME movements. AUs which are annotated less than 5 times are not shown in the table: AU13 was utilized 4 times; AU19, AU23 AU31 and AU34 were used 2 times; AU8, AU21 and AU56 were mentioned just 1 time. For better understanding, the FACS table in Appendix A lists the AUs and their descriptions.

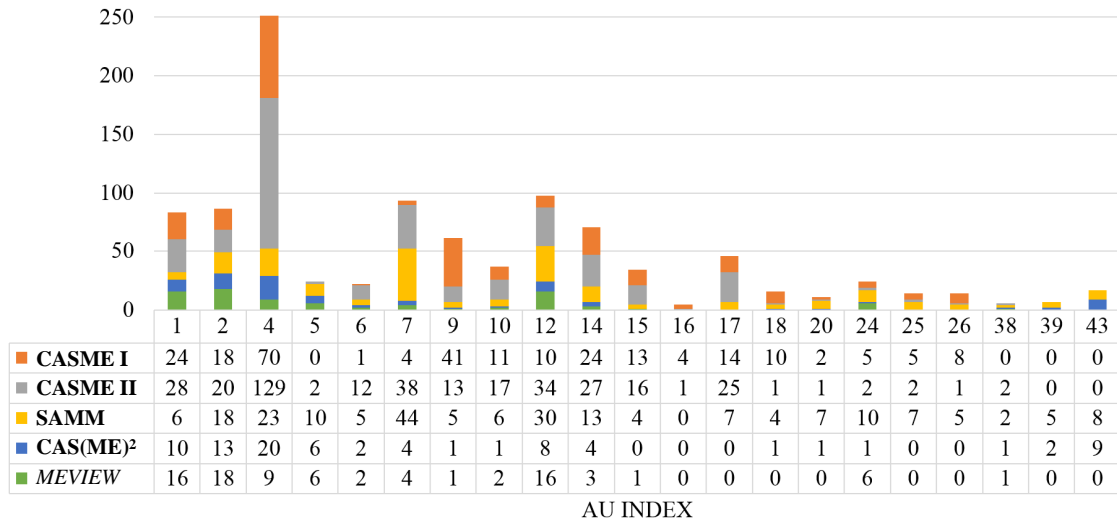


Figure 2-5: Histogram of action units (AUs) annotations for ME databases. The AU amount on the region of eyebrows (e.g. AU 1,2,4) and mouth (e.g. AU 12, 14) indicate that these two regions have the most frequent ME movement.

Davision et al. [21] proposed an objective ME classification method using AUs. The facial movements are labeled by AU combinations, and this would help to avoid the uncertainty cause by subjective annotation. In the Facial Micro-Expression Grand Challenge (MEGC) organized by FG2018 [139], the ME classes for recognition are labeled by this objective classification.

Emotional labels (A.2) are used for ME recognition. As listed in Table 2.7, almost all the ME datasets have emotional labels except Canal9 [112] and Polikovskiy's Database [101]. As mentioned in paragraph *Spontaneous Micro-Expressions* of sub-section 2.1.2, the emotion labels differ in different databases. However, there are two exceptions. One is the Silesian Deception Database [105], who contains micro-tension, eye closures and gaze aversion of subjects. Another one is Canal9, there are no emotional labels, the samples

are annotated by agree/disagree since the videos came from debate scenes. Until now, to our knowledge, there are no ME databases that provide facial features or emotional dimensions. Table 2.9 shows a quantitative summary for commonly used databases, and they list the emotion classes and the numbers of corresponding samples. SMIC-sub and CAS(ME)<sup>2</sup> are shown in both tables, because they have the two types of emotional labels. In SMIC-sub, positive contains all the samples annotated with happiness; negative is the ensemble of disgust and sad samples; fear and surprise samples are not included in the 2 emotional classification. In addition, as the SAMM database is a micro facial movement dataset, there are still 26 video samples which are classified as emotion 'other'.

Table 2.9: Emotion classes and sample numbers for micro-expression databases.

Dataset	Positive	Surprise	Negative	Others
SMIC-sub	17	0	18	0
SMIC-HS	51	43	70	0
SMIC-VIS/NIR	28	20	23	0
SMIC-E-HS	51	42	71	0
CAS(ME) <sup>2</sup>	8	9	21	19

(a) Part 1.

Dataset	H	D	Su	R	T	F	C	Sa	He	Pa	A
S-sub	17	10	20	0	0	16	0	8	0	0	0
CASIA	4	4	7	30	48	1	2	0	0	0	0
CASIB	5	42	14	10	23	1	0	6	0	0	0
CAS <sup>2</sup>	15	16	10	0	0	4	0	1	1	2	0
MEV	6	1	9	0	0	3	7	0	0	0	2
SAMM	26	9	15	0	0	8	12	6	0	0	57
CASII	33	60	25	27	102						

(b) Part 2. H: Happiness, D: Disgust, Su: Surprise, R: Repression, T: Tense, F: Fear, C: Contempt, Sa: Sadness, He: helpless, Pa: pain, A: anger. S-Sub : SMIC-sub, CASIA: CASME I-A , CASIB: CASME I-B, CAS<sup>2</sup>: CAS(ME)<sup>2</sup>, MEV: MEVIEW, CASII: CASME II.

### 2.1.3 Discussion on Micro-Expression Database

The posed ME is a reaction commanded by the brain. The duration is longer than that of spontaneous ME. Yet, the short duration is one of the most important characteristics for ME. Hence, posed datasets are not used anymore.

Nowadays, the majority of automatic ME analysis researches performed their experiments on spontaneous ME databases. CASME II [135] and SMIC [66] are two most commonly used databases. Each spontaneous database has its own advantages. CASME I, CASME II and SAMM have both emotional labels and AU labels. SMIC provides a possibility to analyze ME by multi-modalities. SAMM responds to the necessity of multi-ethnic groups. In addition, SAMM has not only the ME but also other facial movements, and CAS(ME)<sup>2</sup> contains both macro and micro expressions. Moreover, the video length of these two databases is longer than the others. The advantages of SAMM and CAS(ME)<sup>2</sup> can help to enhance the ability to distinguish ME from other facial movements. Thus, they are two very promising databases for improving the ME analyzing performance in the real world.

Nevertheless, there is still plenty of work to do. First of all, the genericity of subjects should be increased.

1. There are too few participants and most of them are university students. The age range needs to be extended. For example, the wrinkle on the face may influence the recognition result. Furthermore, the students do not have much experience of hiding their emotions in high stake situations. To apply the ME analysis in the real world, e.g. interrogation or medical care, it needs to recruit more participants from society.
2. As it's a difficult task for children to hide their genuine emotions, the feature of facial movement could be different from that of adults. Thus, ME samples collected from children should be considered. However, building a database containing children subjects would concern many legislative issues.

Secondly, more modalities, e.g. infrared video, could help improve the recognition ability by cross-modalities analysis.

Thirdly, as the research on automatic MESR just begun in the last ten years, almost all the ME databases were built in a strictly controlled laboratory environment to facilitate the pre-processing. Along with the development of MESR research, in-wild-world ME videos sequences with more occlusions, such as pose variation, hair on the face, lightning change, etc., are expected by collecting from TV shows or by crowd-sourcing.

Fourthly, concerning annotation, utilizing AU annotations could be a more objective way for classification of facial movements. Meanwhile, the accuracy of annotation needs to be improved since there are still many non-labeled detected facial movements in the existing databases.

Fifthly, the number of ME databases could be augmented by considering FACS Coded Databases like DISFA [89] and BP4D [143]. The spontaneous facial expressions in these databases are labelled with AUs intensities, expression sequences with short duration and low intensity could be used as ME samples.

In addition, due to the limited acquisition condition, we are looking forward to a comprehensively collected database by cooperation among worldwide research groups.

The last discussion is about the definitions of eye gaze change, subtle expression and masked expression. They have not attracted much attention. Nevertheless, for the future ME database construction and ME analysis, we think that it is worth discussing.

1. The eye gaze shifts also reveals the personal emotion, even without any action units that associate with it. It could be considered as a clue for identifying MEs. For instance, as we mentioned in the subsection 2.1.2, eye change for eastern people carries emotions. However, to officially use it as ME sample, it still needs acknowledgment from psychologists and automatic MESR research communities. Furthermore, the samples in Silesian Deception Database [105] could be used for analyzing the ME with eye gaze shift.
2. The subtle expression is a small facial movement (spatial), but the duration could be longer than 500ms. The study of subtle expression would be a challenge due to the duration is not defined.
3. Regarding the masked expression, there might be some MEs masked in other facial movements. For example, the tense expression could be hidden during an eye blinking. Analyzing this kind of ME seems to be impossible based on currently proposed methods. We are looking forward to more studies on this problem.

#### **2.1.4 Conclusion for Micro-Expression Databases**

By comprehensively reviewing the existing databases, this section gives some guidelines and suggestions for the further ME database construction. Regarding databases, 13 characteristics are presented in 4 categories for posed, spontaneous and in-the-wild databases. This classification could help other researchers to choose databases as needed. The future direction for databases is under discussion. The diversity of the population and the number of modalities should be increased. More in-the-wild databases are expected.

## 2.2 Result Evaluation Methods and Metrics

In this section, we propose a comprehensive survey, which focuses on result evaluation methods and metrics for ME recognition and spotting respectively. For recognition, besides a quantitative summary for each metric, we also reviewed and discussed the number of recognition classes. Concerning ME spotting, the metrics are introduced based on the different spotting methods. This section also gives a discussion on the metric consistency and its future direction.

The section is organized as follows: subsection 2.2.1 and subsection 2.2.2 review respectively the evaluation methods for ME recognition and spotting. Subsection 2.2.3 concludes this section.

### 2.2.1 Evaluation of Micro-Expression Recognition Result

Micro-expression recognition aims at identifying the emotion type of a ME video sequence. The classification is usually performed by machine learning. There are more than 90 articles on ME recognition. The metrics are the same in most papers. Yet, the number of emotion classes for recognition differs depending on the different databases and the researchers' choice. The following two sub-subsections present the common metrics and discuss the emotion classification for micro-expression recognition.

#### Micro-Expression Recognition Metrics

TP (true positive), FP (false positive), TN (true negative) and FN (false positive) are the basic measures for classification system evaluation [103]. The main metrics for ME classification are accuracy / recognition rate and confusion matrix. Table 2.10 summarizes the amount of 7 metrics in 80 articles, that we reviewed for ME recognition. As shown in this table, accuracy (ACC) is the most commonly used metric for automatic recognition:

$$ACC = \frac{TP + TN}{TP + FP + FN + TN}$$



The result is always evaluated by n-fold cross validation or leave-one-subject-out validation.

Table 2.10: The number and frequency of ME recognition articles, according to the metrics used. CM means the confusion matrix, time includes the training time, recognition time and computation/run time.

Metric	Accuracy	CM	F1-score	Recall	Precision	Time	ROC
# of articles	<b>67</b>	27	17	14	11	6	4
Frequency (%)	<b>83</b>	34	21	17	14	7	2

We classified the published articles depending on their corresponding metrics and databases. The number of articles for each class is shown in Table 2.11. A table that lists all the references of articles with same classification can be found in Appendix B. CASME II [135] and SMIC [66] are two most used databases. Moreover, the number of articles used two new published databases: SAMM [19] and CAS(ME)<sup>2</sup> [104] is still small.

Table 2.11: Summary of number of published articles, with their corresponding metrics and databases. CAS I: CASME I; CAS II: CASME II; SMIC includes SMIC and SMIC-E. ACC: accuracy, CM: confusion matrix.

	CAS I	CAS II	SMIC	SAMM	CAS(ME) <sup>2</sup>
ACC	<b>22</b>	<b>61</b>	<b>37</b>	5	<b>2</b>
CM	13	29	17	3	1
F1-score	5	22	13	<b>6</b>	NaN
Recall	3	13	8	2	NaN
Precision	2	11	7	NaN	NaN
Time	1	5	5	NaN	1
ROC	2	3	2	2	1

As the ME samples correspond to different emotion types, to reveal the classification performance for each emotion class, the confusion matrix is worth to analyze. In the mean time, the emotion sample distribution is not balanced [60], some emotions account for a great proportion in the databases. For instance, 50% samples in CASME I-A [137] are labeled as tense. This situation will influence the accuracy of the entire classification process. Analyzing the confusion matrix would help to improve the ability to distinguish singular emotions. In addition, more and more metrics are used to evaluate the machine learning system, e.g. TPR/recall (True positive rate), FPR (False positive rate), precision, ROC (Re-

ceiver Operator characteristic Curve), etc. One of the most highlight metrics is F1-score:

$$\text{F1-score} = 2 \times \frac{\text{precision} \times \text{recall}}{\text{precision} + \text{recall}} = \frac{2\text{TP}}{2\text{TP} + \text{FP} + \text{FN}}$$

As shown in the formula, F1-score considers both the precision and recall: it is the harmonic average of these two metrics. Thus, F1-score can evaluate the method in an unbalanced dataset. The higher the F1-score is, the more robust the system is.

Moreover, even though the most articles focus on the recognition rate and other above mentioned metrics, there are a few articles which also give the metrics to improve the efficiency of the proposed method. For instance, Hao et al. [41] listed the training time and recognition time. Wang et al. gave the recognition time [124], and other papers ([78, 84, 150] etc.) provided computation time/run-time.

The above metrics can help to evaluate the recognition method in a single database. The amount of ME samples in the published database is smaller than 300, and the majority of recognition methods are trained and tested in one small-scale database. This situation causes problems. In the real world, the data could come from different sources. Hence, to improve the recognition reliability, the cross-database ME recognition is necessary. In [151, 150], Zong et al. performed the cross-database recognition on SMIC [66] and CASME II [135]. Besides, the MEGC (Micro-Expression Grand Challenge) of FG2018 [139] proposed a ME cross-database recognition challenge [56, 98, 90], CASME II and SAMM [19] are utilized for evaluation.

There are two kinds of cross-database recognition. One is to use all the samples from selected databases, and the method is evaluated by the n-fold cross validation. The other one is training the proposed method in one database and then testing the method in another database. For the first kind of cross-database, F1-score can be given for evaluation. Meanwhile, for the second situation, the results could be evaluated by two metrics which have been typically used in cross-database speech emotion recognition: unweighted average recall (UAR) and weighted average recall (WAR). UAR is defined as the average accuracy of each class without consideration of sample numbers in each class; and the WAR is obtained

by dividing the TP sample numbers by the sum of all the samples:

$$UAR = \frac{\text{Accuracy sum of each class}}{\text{Total number of classes}}$$

$$WAR = \frac{\text{Number of TP in all classes}}{\text{All samples}}$$

Evaluating cross-database recognition by these two metrics can help to avoid the unbalanced emotion classes distribution problem.

### **Recognition Classes**

The emotion classes for recognition vary depending on different chosen databases. This is no valance/arousal space for micro-expression recognition. As listed in Table 2.9 of subsection 2.1.2, there are various emotional classifications for some commonly used databases. One type depends on the emotion valence, and the other one type classifies the expression based on 6 basic emotions [29].

Furthermore, since the emotion classes have very different volumes, some authors have defined their own emotion classes. They may combine emotion classes which have small proportions in the entire database into one class. For example, in [131], the emotion classes for CASME I [137] was set as positive, negative, surprise and others. Another solution is to select useful samples for evaluation. For instance, in [37], only ME samples in CASME I corresponding to happiness, surprise, repression and tense are used for the experiment. For articles which performed their experiments on SMIC [66], the most common emotion classification is positive, negative and surprise [67]. Meanwhile, for articles using CASME II [135], the ME samples were usually classified as happiness, surprise, repression, disgust, and others [119].

Table 2.12 lists all kinds of emotional classes and the corresponding number of articles. The emotion classes of database SMIC (positive, negative, surprise) and CASME II (happiness, sadness, disgust, repression, tension/others) are most frequently used. Since there are many combinations, it is difficult to analyze different methods. Hence, researchers could use these two kinds of emotion classes for result comparison.

Moreover, as shown in Table 2.12, it is worth noticing that there are only 5 emotional classifications, which contain neutral emotion. Usually, the researchers use the frames from onset to offset as recognition samples. There are few articles recognizing videos samples who do not contain ME sequences. The frames that are not labeled as ME may contain lots of unrelated facial movements. Thus, it would cause a high false positive for recognition. However, to apply the automatic ME analysis in real life, it requires the ability of spotting the frames that contain ME in a long video. Then these frames can be used for ME classification. We will discuss the ME spotting in section 2.2.2.

Table 2.12: Summary of emotion classes for micro-expression recognition. The two most commonly used emotion classes are highlighted in bold. P: positive, N: negative, H: Happiness, D: Disgust, SU: Surprise, R: Repression, T: Tense, F: Fear, C: Contempt, SA: Sadness.

# of emotions	Emotion types	Article number
2	P, N	2
3	<b>P, N, SU</b>	<b>28</b>
	P, N, Neutral	1
	H, SU, SA	1
4	P, N, SU, others	9
	SU, R, T, D	6
	SU, R, T, H	1
5	Attention, SU, D, R, T	1
	<b>H, SU, D, R, T/others</b>	<b>33</b>
	H, SU, SA, A, Neutral	1
6	H, SU, D, F, SA, A	2
	H, SU, D, F, SA, R	1
	H, SU, D, F, T, Neutral	1
7	H, SU, D, F, SA, A, Neutral	1
	H, SU, D, F, SA, A, C	2
8	H, SU, D, F, SA, A, C, Neutral	1
9	H, SU, D, F, SA, A, C, T, others	1

One main difficulty in ME recognition is to identify precisely the facial movement as one definite emotion without consideration of gesture and context. It occasionally exists some conflicts between emotional label manually annotated by the psychologists and the recognition result performed by machine learning. Hence, the objective classification has been encouraged. It means that classes are built by the combination of AU. This could avoid the above-mentioned conflicts. Recognizing facial movement with AU combination

would be more rational and reliable rather than defining the emotion type. What's more, this kind of classification can serve to unify the number of classes in different databases. It would facilitate the comparison between different methods.

To develop the transferability of ME recognition methods, the experiments on cross-database is necessary. In two published articles of Zong et al. [151, 149], the emotional classes are positive, negative and surprise, and the databases are SMIC [66] and CASME II [135]. Meanwhile, in articles published in FG2018-MEGC [56], the ME recognition was performed on five objective classes. CASME II and SAMM [19] were utilized. The number of recognition classes and the sub-groups vary depending on the chosen databases. It is reasonable to use different emotion classes when the databases are different. Yet, when the chosen databases are the same, we need to pay attention to the fact that the experiments should take the same sub-groups to facilitate the result comparison.

## **2.2.2 Evaluation of Micro-Expression Spotting Result**

ME spotting is a broader term for identifying whether there is a ME in a video or not. In contrast, ME spotting means more specifically locating the frame index of ME in videos. In this chapter, we use spotting to represent these two definitions. There are only around 30 articles for automatic ME spotting, and the metric consistency for result assessment is still an open field. The final analysis result could be evaluated by temporal window or by frame, and the metrics vary according to different methods. Therefore, this situation makes it difficult to compare results obtained by various methods. This subsection firstly presents the result evaluating methods for ME spotting, then introduces the most frequently used databases. Furthermore, the metric standardization is discussed at the end of this subsection for the future research.

### **Analyze Methods and Corresponding Metrics**

There are just a few research articles on ME spotting, less than 40 articles from 2009 to 2019. In 2014, Moilanen et al. [91] published the first article which performed the ME spotting method on spontaneous database. The results were evaluated per video. Except

this article [91], the spotting results from other articles are evaluated by different criteria, i.e. per frame and per temporal window. We discuss the metrics for these two evaluation criteria in the following paragraphs.

***Micro-Expression Spotting per Frame*** ME per frame means spotting ME frames in a video sequence. There are two types of spotting result: one is finding the apex or onset frame [77] (DF.1), one is locating the frames in ME interval [23]. As presented in [63], the second one contains two kinds of spotting methods, i.e. feature difference (DF.2) and machine learning (DF.3).

Table 2.13 lists the published micro-expression spotting methods per frame and their corresponding metrics. Some methods are re-implemented in several articles and they are evaluated by different metrics. For instance, the original article [67], which proposed the LBP- $\chi^2$  distance method, used accuracy and ROC. Meanwhile, Li et al. [63] compared their proposed method with the LBP- $\chi^2$  distance method. In this paper, the results are evaluated by TPR and F1-score. It is an existing problem for micro-expression spotting research. In order to be able to compare the results, researchers need to re-produce the methods in published articles.

For the first one (DF.1), the average error (AE), mean absolute error (MAE) and standard deviation (STD) are used as the evaluation metrics. Liong et al. [77] introduced another metric: Apex Spotting Rate (ASR). As it is not yet commonly used by other research groups, this metric is not listed in Table 2.13.

The second spotting format per frame is the most common ME spotting method, and the basic measure is TPR:

$$\text{TPR} = \frac{\text{all detected true positive frames}}{\text{sum of all ME frames in each video sequence}}$$

For feature difference methods (DF.2), ROC and AUC (Area under the curve) are two popular metrics as there are some thresholds to be adapted [91]. The ROC is drawn as TPR versus FPR. The larger the AUC, the better the system performs. However, the ROC does not deal well with the unbalanced sample distribution situation. We take an example of a

Table 2.13: Published spontaneous micro-expression spotting per frame methods and their corresponding general metrics. The method highlighted in bold is the most commonly used method for comparison. DF.1: apex/onset spotting methods, DF.2 : feature difference methods, DF.3: machine learning methods. ACC: accuracy; ROC: receiver operating characteristic curve; TPR: true positive rate; AE: average error, and this column also includes the articles which used MAE (mean absolute error).

	Methods	ACC	ROC	TPR	F1-score	AE / MAE
DF.1	OS [79]					✓
	CLM [134]					✓
	LBP-correlation [134]					✓
	OF [134]					✓
	Spatio-temporal integration of OF [96]	✓				✓
	RHOOF [87]					✓
	Apex frame spotting by frequency filter [69]					✓
DF.2	<b>LBP-<math>\chi^2</math> feature difference</b> [91, 67, 79, 20]	✓	✓	✓	✓	✓
	HOOF [67, 20]	✓	✓	✓	✓	
	MDMD [116]	✓				
	HOG- $\chi^2$ feature difference [23]			✓	✓	
	3D-HOG- $\chi^2$ [20]	✓	✓	✓	✓	
	IP [83]		✓			
	OF [83]		✓			
	Riesz Pyramid [24]		✓			
CFD [40]		✓				
DF.3	Adaboost [130]		✓			
	Characteristic image intensity [52]		✓			
	Motion descriptors[13]	✓				
	LTP-SVM [63]	✓		✓	✓	

fictional database, in which there are 100 false samples and 10 true samples. Supposing that the system detects 20 FP samples and 9 TP samples, the point position of (FPR, TPR) on ROC would be (0.2, 0.9). If we just consider the ROC metric, the system performs well. However, the fact is that there are too many false positives influencing the reliability of test results. ROC and AUC are useful to study the parameter influence for one method. Yet, they are not suitable for result comparison among different methods. F1-score is commanded because it considers both TPR and FPR, and deals well the problem of imbalanced sample distribution, as introduced in subsection 2.2.1.

Meanwhile, machine learning methods (DF.3) for micro-expression apply the same evaluation method as most classification methods, i.e. they use the most common metrics for classification: recall, precision, confusion matrix, and F1-score [13].

Another key point for the measurement per frame is the ground truth setting. The ground truth can be the interval from onset to offset [23] or  $[\text{onset} - k/2, \text{offset} + k/2]$ , where  $k$  is the sliding window length in spotting method [91]. All these varieties of the result calculation would cause the problems for comparison.

***Micro-Expression Spotting per Window*** In 2017, Tran et al. [111] proposed a novel spotting method. The spotting is evaluated by a search window  $W$ . The window would be considered as positive if it meets the following condition:

$$\frac{GT \cap X_W}{GT \cup X_W} \geq \epsilon$$

where  $GT$  means ground truth in the video, i.e. the frame index from onset to offset;  $X_W$  is the spotted frame index in the search window,  $\epsilon$  is set to 0.5 according to the study in [130]. And if the spotting result is smaller than 0.5, it is treated as negative. Thus, each searching window has one corresponding label (positive or negative). Then these windows are used as training and testing samples for machine learning. To analyze the performance, the Detection Error Tradeoff (DET) curve is firstly used. The measurement DET is plotted as: miss rate versus FP per window.

This kind of measurement can help to reduce the true negatives which are caused by



spotting per frame. Moreover, the labels of frame samples are sometimes not quite accurate, and it would influence the result evaluation. Spotting and evaluating the result per window could help to avoid this situation. There is a disadvantage for this evaluation per interval is that for each spotting method, the interval length might be different, then the result could not be compared between methods.

Even though there are a few articles which spot ME per temporal window until now, we think this method is promising and it could be extended to other general metrics or methods.

### The Frequently Used Databases for Micro-Expression Spotting

In this sub-section, we will introduce the spontaneous ME database for ME spotting. Because there is only one article [52] which spotted micro-expression in in-the-wild database MEVIEW. All the other articles performed their methods on spontaneous database. Table 2.14 shows the number and frequency of databases used for spotting. CASME II and SMIC-E are two databases used most frequently.

Table 2.14: Database numbers and frequency (%) for ME spotting. The number of two most frequently used databases are highlighted in bold. CAS II : CASME II, CAS I ; CASME I.

Databases	CAS II	SMIC-E	CAS I	SAMM	CAS(ME) <sup>2</sup>	MEVIEW
# of articles	<b>12</b>	<b>11</b>	6	2	2	1
Frequency (%)	<b>57</b>	<b>52</b>	29	9	9	5

In addition, the two most recent spontaneous databases (SAMM [19] and CAS(ME)<sup>2</sup> [104]) are recommended. They contain longer video samples compared with other spontaneous databases. There are more non-ME samples, including neutral faces, eye blinks, eye gazes change, subtle head movement, etc. As applications in real life are applied on long videos, performing experiments on these two databases would help to improve the adaptability of spotting methods.

## Standardization of Micro-Expression Spotting Metrics

All the researchers try to provide as many metrics as possible in the article to analyze comprehensively the method. The used metrics seem to be adapted to the proposed method. As the methods are not the same, each paper uses different metrics from the other papers. As shown in Table 2.15, all the frequencies are less than 50%, there is no dominant metric for spotting result evaluation. MAE is used for single frame spotting, ROC works well when there is an important parameter to manipulate, while ACC is the basic metric for classification. Hence, it's difficult to define which result evaluation method or metric is better.

Table 2.15: The number and frequency of ME spotting articles, according to the metrics used. ACC: accuracy, AE: average error, AE also includes MAE.

Metric	ROC	ACC	AE	TPR	F1-score
# of articles	<b>9</b>	<b>7</b>	<b>7</b>	3	3
Frequency (%)	<b>43</b>	<b>33</b>	<b>33</b>	14	14

Here are some suggestions for the future ME spotting. As the ME is a continuous facial movement, spotting ME frames by sliding window and evaluating the result per windows would help to provide more reliable ME sample frame for the further recognition process. Besides, as shown in [63], the movement variation of ME is more regular during the period of the onset compared to the variation after apex frame. Spotting onset frame or the onset-apex interval would obtain a more accurate result.

In addition, concerning the metrics, F1-score is recommended. There are three reasons:

- First of all, as the machine learning method is the trend for ME spotting, F1-score is a very important metric for demonstrating the performance of machine learning process.
- Secondly, the ratio of ME frames and non-ME frames is quite small in a long video sequence. The F1-score can avoid the unbalanced sample distribution problem with consideration of TP, FP and FN.
- Thirdly, due to the local and brief nature of ME, it is challenging for a spotting system to distinguish the ME movement from other facial movements. For example,

the eye blinks, the eye gaze and the subtle head poses change would cause many false positives. The problem can be reflected in the value of F1-score while the value of TPR or accuracy could be high.

MCC (Matthews correlation coefficient) [88] could also be used for the binary classification even if the classes have different sizes. MCC is presented:

$$MCC = \frac{TP \times TN - FP \times FN}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}}$$

It is regarded as a balanced metric as it considers all four basic metrics (TP, FP, TN, FN). The value varies between  $[-1, 1]$ . The ideal situation is that MCC equals to 1. 0 means that the classification result has no difference with random prediction. And -1 means the result is worse than that of random prediction. The MCC has already been a common metric for the community of automatic facial expression analysis. We could expect it would be a useful metric for the result evaluation of automatic ME analysis.

### 2.2.3 Conclusion for Result Evaluation Methods and Metrics

Detailed quantitative information of the metrics and emotional classes for micro-expression recognition help researcher to establish a proper result evaluation method for their own research purpose. In addition, objective classes are recommended to obtain more rational micro-expression recognition result. Concerning micro-expression spotting, we pointed out that the consistency of spotting metric because of the difficulty of result comparison. The ME spotting per interval is recommended. To unify the metrics for algorithm comparison, F1-score seems to be adequate, for it considers both precision and recall. Due to the limited number of articles on ME spotting, it is not easy to find a direction for metric consistency. We are looking forward to more research on micro-expression spotting, which would help to standardize the result evaluation methods.

According to our above survey, we have chosen to perform experiments on CASME I and CASME II for my thesis research (micro-expression spotting), since most articles used these two databases to evaluate the result. Furthermore, SAMM and CAS(ME)<sup>2</sup> are used to test our method in long videos. Meanwhile, the F1-score per frame is applied for the spotting result evaluation. In addition, we have developed a new result evaluation method, which is inspired by result evaluation per interval.

## 2.3 Micro-Expression Spotting Methods

Our research focus on micro-expression spotting. The review of related works indicates the merits and shortness of current spotting methods. Since the micro-expression is an involuntary facial expression, we focus on the ME spotting methods which are developed based on spontaneous or in-the-wild databases. [107] used a 3D-gradient descriptor to detect micro-expression in USF-HD database. [102] applied the micro-expression detection in hi-speed video based on facial action coding system (FACS) using the histogram of 3D-gradient features. These two articles are not discussed in the following paragraphs because the methods are performed on posed micro-expression databases.

Subsection 2.3.1 compares the related method depending on their algorithms, then Subsection 2.3.2 investigates the features for micro-expression spotting. Subsection 2.3.3 concludes this section and enlightens the direction of our research.

### 2.3.1 Related Works Comparison by Algorithms

The common workflow for micro-expression spotting is to firstly extract feature and then to spot ME based on proposed algorithms. In this subsection, we only pay attention to the algorithms for ME spotting after feature extraction, i.e. regardless the methods to extract features. The two main algorithms for ME spotting: feature difference and machine learning are introduced in following paragraphs.

**Feature Difference Methods** Most methods utilize the feature difference. The aim is to calculate the difference between dedicated features in a time window. The most significant movement is spotted by setting a threshold in the entire video.

Moilanen et al. [91, 65] used the linear binary pattern (LBP) feature difference analysis to spot ME. MEs were then extracted by thresholding and peak spotting. Yan et al. [135] quantified ME and spotted the apex frames by three feature extraction algorithms: Constraint Local Model (CLM), LBP and optical flow. Liong et al. [77] developed the method and spotted the apex frame by employing a binary search strategy. Patel et al. [96] utilized optical flow and then a spatiotemporal integration to spot apex frame and identify

the location of onset and offset. Davison et al. [20] applied 3D-HOG as the feature distance measure to calculate the dissimilarity between frames and detected the ME using an individualized baseline. Liong et al. [74] spotted apex frame by employing optical strain which is more effective in identifying the subtle deformable facial muscle. Li et al. [68] integrated a deep multi-task learning method to locate the facial landmarks and then spotted the ME with HOOOF (histograms of oriented optical flow). Yet, deep learning is only applied for feature extraction, the algorithm for ME spotting is still feature difference in this article. Wang et al. [115] proposed the main directional maximal difference analysis of optical flow to spot the ME. Lu et al. [83] presented a method with a low computation cost based on differences in the Integral Projection (IP) of sequential frames for ME spotting. Ma et al. [87] proposed a region histogram of oriented optical flow (RFOOF) feature to spot the apex frame. The feature difference is calculated per Region of interest (ROI) and the spotting result can be obtained by a spatial fusion. Inspired by the video magnification, Duque et al. [24] extracted features by Riesz pyramid. A filtering and masking scheme was applied to segment the interested motion, including eye blinks. Han et al. [40] introduced a collaborative feature difference method which combined the LTP and MDMO. In addition, the Fisher linear discriminant was used to assign a weight for each ROI. Li et al. [69] detected apex frames in frequency domain, depending on the correlated relationship between apex frame and the amplitude change in the frequency domain.

Since micro-expression is almost undetectable by one single frame, the main advantage of these approaches is to be able to make comparisons between frames over a time window of the size of ME. Yet, only the first and last frame in the interval are utilized for the feature difference calculation of the current frame. They do not take into account the temporal variation of ME. Another shortcoming of feature difference method is that, they spot the movement between frames, and not specifically the ME movement. There are two articles [20, 115] which have made improvements: Davison et al. [20] created an individualized baseline to differentiate the videos which do not contain the micro-expression, [115] distinguishes the macro-and micro expression by their duration. Nonetheless, the ability to distinguish MEs from other movements (such as blinking or head movements) remains weak, especially in long videos. Feature difference method would cause many false

positives as it spot the movements which are above the threshold, regardless of whether or not they are ME.

**Machine Learning Methods** Nowadays, methods utilizing machine learning are emerging. Xia et al. [130] utilized Adaboost model to estimate the initial probability for each frame and then a random walk model to spot the ME by considering the correlation between frames. Hong et al. [111] proposed a multi-scale sliding window based approach. LBP-TOP, HOG-TOP and HIGO-TOP were extracted as the feature and MEs were detected by binary classification. Borza et al. [13] used the movement magnitude across frames by simple absolute image differences, then the Adaboost algorithm was applied to detect ME frames. Husák et al. [52] tried to spot micro-expression in an in-the-wild database. The feature was extracted based on analyzing image intensity over a registered face sequence. Then an SVM classifier was used to for the classification. Furthermore, [144] employed CNN for the first time to perform the ME spotting.

Since the features extracted from ME are trained for the classification, the machine learning process enhances the ability of distinguishing ME from other facial movement. Yet, there are less than 10 papers using machine learning or deep learning for ME spotting. The ME spotting in this domain is limited by the size of the database. The amount of ME samples in published databases is not large enough to train a performant classifier. In addition, the performance highly depends on the relevancy of the features. Features adapted to characteristics of ME need to be exploited.

### 2.3.2 Related Works Comparison by Features

Various features are applied to spot micro-expressions, including LBP [65], HOG [20], optical flow [96, 68, 115, 40], optical stain [74], integral projection [83], features extracted by Riesz pyramid [24] and features extracted in frequency domain [69].

ME is a local facial movement with a very short duration. Its local and temporal characteristics can help to spot ME in videos.

- Concerning the local information, facial movements such as eye blinks may have a similar feature compared with ME, but the regions where the movement occurs

are different. Local characteristic is useful to enhance the ability to differentiate the ME from other movements, which have similar duration and intensity. Yet, the local information has not attracted sufficient attention in ME spotting research. For almost all the methods, even if the features are extracted from ROIs, an early spatial fusion is performed on all these extracted features to obtain a global descriptor for one entire frame, which is the input of the algorithms. [87] and [40] are two exceptions: the first one calculated the feature difference per region of interest (ROI), and the second one assigned a weight for each ROI.

- Regarding the temporal information, features like LBP-TOP [111] extracts temporal characteristics in a small temporal window. Yet, the duration is too short to represent a whole temporal movement pattern for ME. And as introduced in above subsection, the temporal information is taken into account in algorithms, but not in the feature construction of ME. Focusing on the temporal pattern variation in a ME duration may help improve the result.

### 2.3.3 Conclusion

Compared with feature difference method, machine learning seems to be the more performant algorithms for micro-expression spotting. In addition, the local information and the temporal pattern are two important characteristics of micro-expression. Exploiting them as features for micro-expression spotting may help improve the performance.

As a conclusion, the temporal and the local information is taken into account for the feature extraction in our proposed method. Meantime, the machine learning method for ME spotting is considered to improves the ability to distinguish ME from other movements.
--



## 2.4 Micro-Expression Spot-and-Recognize Schemes

In this section, we would like to discuss the micro-expression spot-and-recognize schemes. As the automatic micro-expression analysis are expected to be applied in real life, an entire process need to be considered, in which, the video sequence is the input and finally the emotion class is the output. To our knowledge, there are two kinds of schemas as shown in Figure 2-6. One is to treat the non-micro-expression as one emotional classes, then applying the recognition method to classify the samples into different emotional classes, e.g. [129]. The other one is firstly spotting the micro-expression sequences in long video, then identifying the emotion type of this ME sequence by recognition methods, e.g. [65].

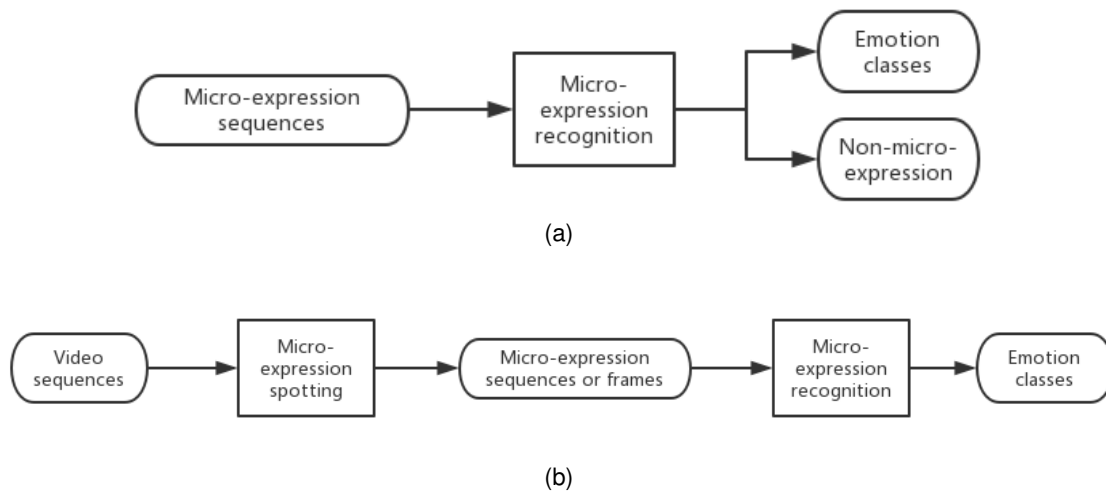


Figure 2-6: Micro-expression spot-and-recognize scheme. 2-6a: the non-micro-expressions are identified by recognition method. 2-6b: the micro-expression samples are firstly spotted in long videos, then they are classified into different emotion classes by recognition methods.

### Non-Micro-Expression as One Emotion Class for ME Recognition

As mentioned in section 2.2.1, there are few recognition methods which could also identify the non-ME expression frames. In these articles, the classes for recognition not only includes the different emotional types but also the neutral expression. The process pipeline is shown in Figure 2-6a. Wu et al. [129] classified the video samples into 6 basic emotional classes and a neutral class. The feature was extracted by Gabor filter and then the GentleSVM was applied for recognition. Lu et al. [85] used a Delaunay-based temporal

coding model for micro-expression recognition, and the experiments included separating micro-expressions from non-micro-expressions. Takalkar and Xu [110] generated extensive datasets of synthetic images using data augmentation and then constructed a satisfactory CNN-based micro-expression recognizer. The classes also contain the neutral expression. Guo et al. [35] proposed a multi-modalities CNNs based on visual and geometric information for the micro-emotion recognition. Yet, the test dataset is iCV-MEFED [86] which is not a commonly used database. In this dataset, samples are classified into 7 emotional classes and a neutral class.

The above mentioned methods regard non-micro-expressions as neutral emotion. However, in the real world, the non-micro-expression is undefined, it could also include many other facial movements, such as macro-expression, eye blinks, slight head movement and etc. Identifying the non-micro-expression as one kind of emotion type is quite challenging, since this task need an accurate definition and annotation for non-micro-expressions. Moreover, for the real application, there might not exist any micro-expressions in the testing video. Thus, this kind of scheme is limited to the videos which contain micro-expressions.

### **Micro-Expression Spot-then-Recognize Scheme**

For the second scheme, the micro-expression samples are firstly separated from non-micro-expression in long videos, then these spotted samples are used for micro-expression recognition. The pipeline is illustrated in Figure 2-6b.

Li et al. [65] proposed micro-expression spotting and recognition methods respectively. As already introduced in subsection 2.3, micro-expressions are spotted by feature (LBP and HOOB) difference method. For recognition task, the LTP-TOP, HOG-TOP and HIGO-TOP were extracted from ME samples. Then a linear support vector machine (LSVM) was used for classification. The recognition experiments were also performed on the spotted ME sequences. The accuracy is lower than the recognition result of ground-truth micro-expression sequences. This is because the spotting process cannot always locate the ME sequences precisely. Hence, it is important to concentrate on micro-expression spotting research and try to improve its performance. In 2018, Li et al [69] detected the apex frame using frequency filter then recognize the emotion type of this apex frame by deep convolution

neural network (DCNN). In the same year, Liong et al [72] proposed an OFF-ApexNet for micro-expression recognition. Apex frame locations are already given in CASME II [135] and SAMM [19]. However, in SMIC [66], just onset and offset locations are indicated. Thus, in this paper, apex frames of SMIC were acquired by calculating the correlation coefficient of LBP feature between frames [77]. Boza et al. [15] also utilized CNN to detect and recognize micro-expressions. The first, the middle and the last frames of a sliding temporal window are used as input volume. At the spotting stage, the intervals are classified as non-ME or ME, then the intervals which are identified as ME went into the recognition process. Compared with the first scheme, spot-then-recognize seems to be easier to adapt to the real situation. The recognition process would only be executed when there are spotted micro-expressions. In addition, there are already many recognition researches for ME samples between onset and offset. Improving the spotting accuracy would be more efficient than classifying the non-micro-expressions into one class.

## 2.5 Conclusion

The automatic micro-expression analysis has emerged in the last decades, and most of the researches are focusing on micro-expression recognition. There is still not much research on micro-expression spotting, which is the bottleneck of automatic micro-expression analysis. In addition, the number of databases that can be used for micro-expression spotting is still limited, and the result evaluation method for spotting also need to be unified. Yet, for the applications in real life, spotting micro-expression in long videos is the first step for micro-expression analysis. It is essential to spot the micro-expression sequences precisely.

Thus, in my thesis, we focus on developing micro-expression spotting method.

- We address this problem by using dedicated local and temporal pattern (LTP) of facial movement. In our system, with the purpose of improving the spotting accuracy, temporal local features are generated from the video in a sliding window of 300ms (mean duration of a ME). This pattern represents the main movement extracted on the time axis by PCA. For all micro-expressions, the local temporal patterns are the same (S-pattern). Using a classical classification algorithm (SVM), S-patterns are then distinguished from other LTP patterns.
- Our method allows to distinguish micro-expression from other facial movements, such as eye blinks and head movements, thanks to a combination of local and global analysis. Movements which are similar to micro-expression (S-pattern) are classified on the local regions. Then, in order to eliminate the false positives, a spatial and temporal fusion analysis is applied from local to global.
- To increase the size of training dataset, we utilize Hammerstein models to synthesize S-patterns. In addition, the model can also filter outliers by learning the patterns of the facial ME movement both in space and duration. This can enlarge the data volume for training while maintaining the ability to differentiate facial movements.

In the following chapters, the method and the experiments will be introduced.



## Chapter 3

# Local temporal pattern for Micro-expression Spotting

As introduced in Chapter 2, local information and temporal variation during a micro-expression may help to improve micro-expression spotting performance. In this chapter, we propose a novel relevant feature for micro-expression spotting: local temporal pattern (LTP). Furthermore, to enhance the ability of distinguishing micro-expression from other movements, the machine learning method is utilized to classify the proposed feature on local regions. A late spatial-and-temporal fusion is then performed to obtain the final micro-expression spotting result. The proposed method consists of three parts: a pre-processing to precisely detect facial landmarks and extract the regions of interest (ROIs), then the computation of local temporal pattern (LTP) on these ROIs and eventually the spotting of micro-expressions. Figure 3-1 displays the overall process.

The chapter is organized as follows: the first three sections present the sub-steps of our method applied in short videos: pre-processing, feature extraction (LTP computation) and micro-expression spotting (classification and fusion). Our method is then adapted to the situations of long videos in section 3.4. Finally, section 3.5 concludes the chapter and points out the requirements to improve the performance of our method.

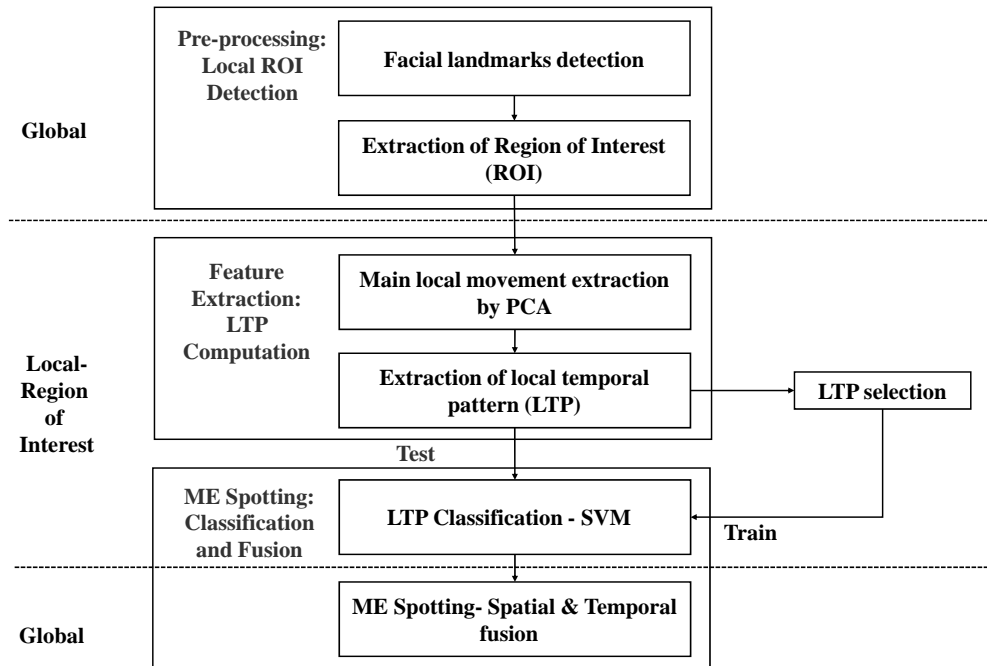


Figure 3-1: Overview of our method. The proposed method contains three steps of processing: pre-processing, feature extraction and micro-expression spotting. We mix both global and local processes. Both sub-step of feature extraction and the first sub-step of micro-expression spotting are performed on relevant regions of interest (ROIs). LTPs including S-patterns are then used as the training samples to build the machine learning model (SVM) for classification. Especially, a final spatial and temporal fusion is performed to eliminate the false positives such as eye blinks. The specificity of the process is the use of local temporal patterns (LTP), which are relevant for micro-expression spotting: micro-expressions are brief and local movements.

## 3.1 Pre-Processing

As the micro-expression is a local facial movement, the analysis on local region allows to extract features which are more relevant to micro-expression. The pre-processing is performed on the face to determine local regions of interest (ROIs). This process contains two stages: facial landmarks are firstly detected, and points which are related to micro-expressions are then chosen to extract regions of interest (ROIs).

### 3.1.1 Facial Landmarks Detection

The first step consists in detecting 49 landmarks (LMs) on the human face for each image. We use the tool 'Intraface' [132]. Except the Section 3.4 which applies the method on long videos, our spotting method is performed on short spontaneous micro-expression video samples (less than 2s). As introduced in Section 2.1.2, the participants barely move and they face directly the camera in the recorded videos. There are few disturbances caused by head global movement. Meantime, the purpose of our method is to extract the deformation of local texture instead that of landmarks. Thus, we just use the detected landmarks in the first frame as reference. Thus. the ROIs in the following frames are extracted depending the landmarks detected on the first frame in the short video.

Regarding the long videos, since the head movement is inevitable for a long period of time, the facial landmarks of each frame are tracked. The long video is divided into several short sequences by a sliding temporal window. ROIs in short sequence are then extracted depending on the landmarks of the first frame in this interval. The detailed process for long video is presented in Section 3.4.

### 3.1.2 Extraction of ROIs

The second step consists in extracting ROIs where the micro-expressions could possibly occur. These ROIs are generated in the form of a square around the chosen landmarks. The length of the side of the square  $a$  is determined by the distance  $L$  between the distance between the left and right inner corners of eyes:  $a = (1/5) \times L$ . Figure 3-2 illustrates the result of pre-processing on an image. ROIs are labeled with the index of the corresponding



landmark. The ROIs include the regions of the two eyebrows and the contour of the mouth.

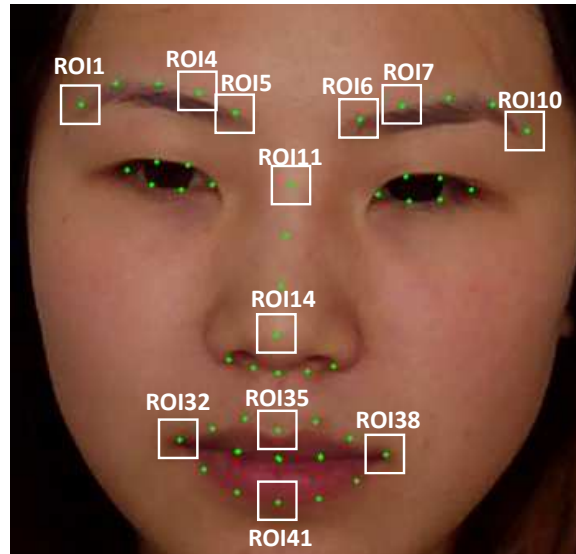


Figure 3-2: Facial landmarks and ROIs distribution. 49 landmarks are detected and ROIs are generated depending on the position of 12 chosen landmarks. 12 ROIs, relevant for micro-expression, are selected from the region of eyebrows, nose and mouth contour.(©Xiaolan Fu)

As presented in Figure 2-5, these ROIs contain most evident action units (AUs) for micro-expression description. The eye area is neglected due to blinking. Because of the rigidity of the nose, this area is chosen as a reference to eliminate overall movements of the head. Table 3.1 gives the link between AU and ROI location.

Table 3.1: Chosen ROIs and related AU index. The ROI index is annotated depending on the detected facial landmarks. 12 ROIs mean that the local regions which have most evident motion for micro-expressions are selected.

Facial region	Related AU	12 ROI index
Eyebrows	1, 2, 4	1, 4, 5, 6, 7, 10
Nose	NaN	11, 14
Mouth	10, 12, 14, 15, 17, 25	32, 35, 38, 41

## 3.2 Feature extraction: Local Temporal Pattern Computation

The aim of this section is to extract a new relevant feature for micro-expression spotting: Local Temporal Patterns (LTPs). As micro-expressions are brief local movements, the LTPs aim at extracting the local information of the texture distortion in a time window of the size of a micro-expression (300ms).

The main local movements extracted by PCA are presented in subsection 3.2.1, then subsection 3.2.2 presents the deduction process of the local temporal patterns. Finally, subsection 3.2.3 demonstrates the unique pattern of LTP for all micro-expressions.

### 3.2.1 Main Local Movement Extraction by PCA

This section presents the process of main local movement extraction by PCA. LTPs are computed for each frame and each ROI. They are based on the change in the grey level texture of the ROI. To detect the main distortion of grey level texture of one ROI though time, we use PCA [6] on the whole ROI sequence. Figure 3-3 illustrates this processing on one of the ROI sequences.

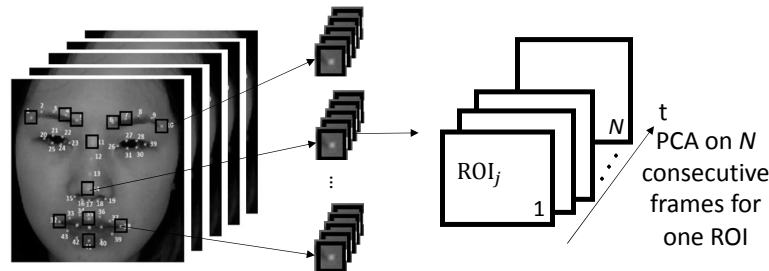


Figure 3-3: PCA on time axis per ROI. A local video sequence of ROI<sub>j</sub> with  $N$  frames (video duration  $\leq 3$ s) is processed by the PCA on the time axis. The first components of PCA conserve the principal movement of grey level texture on this ROI in this duration ( $N$  frames). The video sample comes from CASME I (©Xiaolan Fu)

Let  $N$  be the number of frames in a video, and  $a^2$  be the size of the  $j$ th ROI in pixels, the size of matrix  $I_j$  processed by PCA is  $a^2 \times N$ . We note  $\bar{I}_j \in M_{a^2, N}(\mathbb{R})$  the mean value matrix of each pixel in chosen ROI and  $\Phi \in M_{2, N}(\mathbb{R})$  the projection matrix in the PCA

space reduced to the first 2 dimensions, which conserve more than 80% of the energy. Figure 3-4 shows an example of PCA energy analysis, the first two components conserve more than 80% of the energy.

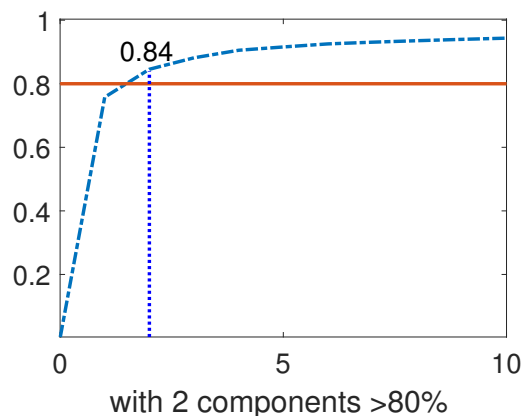


Figure 3-4: PCA energy analysis. The movement is contained in the first 2 components with more than 80% energy. (Sub01\_EP03\_5\_ROI10 of CASMEI)

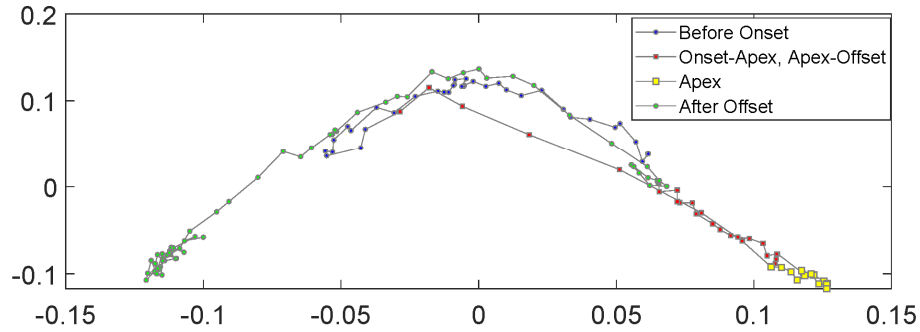
The matrix  $\mathbf{P}_j$  in size of  $2 \times N$ , i.e. the projection of  $I_j$  in the PCA space with the first two dimensions, is obtained by following formula:

$$\begin{bmatrix} P_1^j(x) & \cdots & P_N^j(x) \\ P_1^j(y) & \cdots & P_N^j(y) \end{bmatrix} = \Phi \times \left( \begin{bmatrix} I_1^j(1) & \cdots & I_N^j(1) \\ \vdots & & \vdots \\ I_1^j(a^2) & \cdots & I_N^j(a^2) \end{bmatrix} - \bar{I}_j \right) \quad (3.1)$$

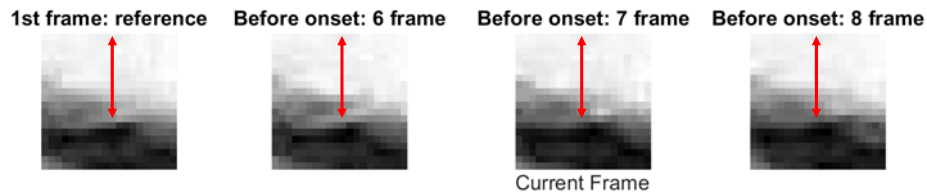
Each point  $P_n^j$  represents the most significant regional motion for one frame. Facial changes can be analyzed on the time axis.

The results of the PCA on a ROI video sequence (ROI 5) of 2.5 seconds (148 frames in 60 fps (Frame Per Second)) are shown in Figure 3-5. It is an interface to observe the global distribution of 2D frames (3-5a) and to compare the image change between the studied frame (current frame) and the first frame of the video. We can find the distance between the yellow point (apex frame) and the first blue point (first frame in ROI sequence) is larger than the distance between the blue points (before onset frames). The movement of ROI are then displayed on grey level images (3-5b, 3-5c and 3-5d), including the first frame, current frame (CF), and the two frames before and after the CF. The red arrow

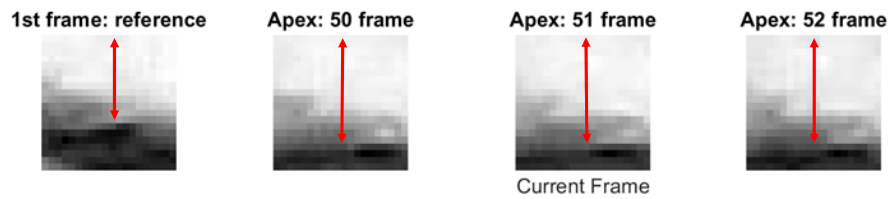
indicates the movement magnitude in the ROI. **By comparing the length of red arrow, a relation between the distance of the points and the magnitude of the movement can be observed: while the distance gets larger, the magnitude increases. That is why we use the distance for micro-expression movement analysis.**



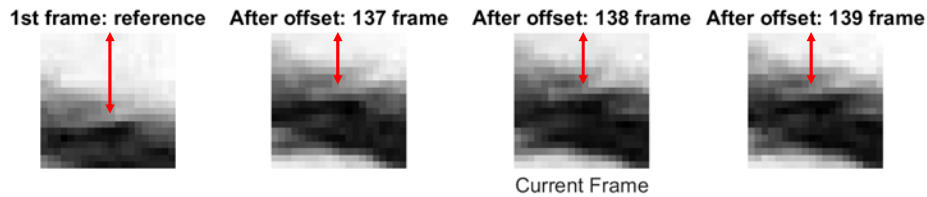
(a) Example of frame distribution after PCA for ROI 5 (Sub01\_EP05\_2 in CASME I)



(b) Comparison of ROI images between the first frame and other frames before onset.



(c) Comparison of ROI images between the first frame and apex frames of ME.



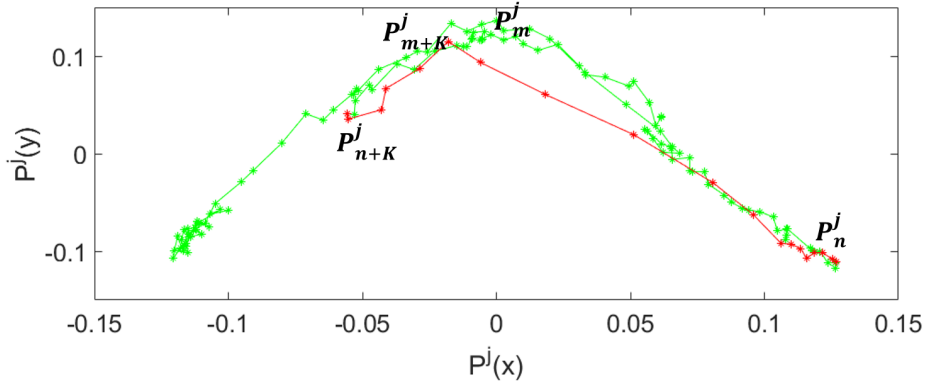
(d) Comparison of ROI images between the first frame and other frames after offset.

Figure 3-5: Interface of the result distribution after PCA process. (3-5a) The point distribution corresponds to all frames in one ROI sequence in PCA projection space. The chosen ROI is the inner side of left eye brow. The blue, red and green dots represent the frames before onset, from onset to offset and after offset respectively, and the yellow dots mean the apex frames. (3-5b, 3-5c and 3-5d) Displacement comparison by ROI images. The first image illustrate the first frame in the ROI sequence. The three images on the right represent the current frame (CF), CF-1 and CF+1. The red arrow shows the displacement of the eyebrow. In 3-5b, there is barely no movements at the beginning of the sequence. In 3-5c, the eyebrow goes down due to the micro-expression. In 3-5d, the eyebrow is raised up compared with apex frame because the ME fades out. Besides, the position of eyebrow is even higher than the first frame due to other facial movements. Comparing the arrow length in the different frames, a conclusion can be obtained, i.e. geometric features in the 3-5a depending on the distribution of frames can represent the temporal displacement in ROI.

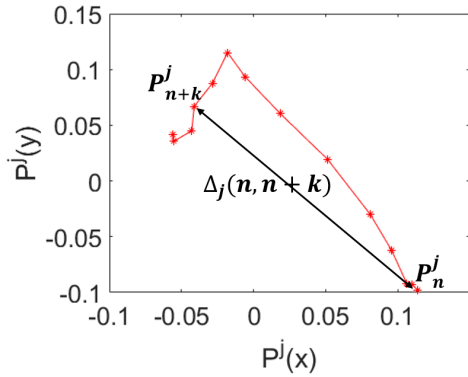
### 3.2.2 Extraction of Local Temporal Pattern

After obtaining frame distribution of ROI sequence after dimension reduction, the feature of LTP is processed by distance calculation and normalization.

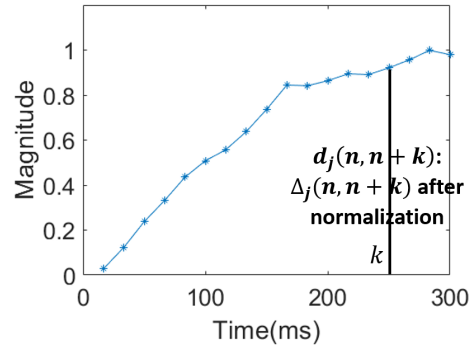
**In this section, a point is not the x,y coordinates of a pixel in the image, but the grey-level value of a ROI.**  $P_n^j$  represents the projection of  $n$ th frame for ROI $_j$  on PCA space with the first two components:  $P_n^j(x), P_n^j(y)$ . The temporal trajectory of the points  $P_n^j$  gives the information of the local texture movement in the ROI. A relation between the distance of the points and the movement magnitude can be found: while the distance gets larger, the magnitude increases. Figure 3-6a shows an example of the trajectory of the points  $P_n^j$  for ROI $_j$  of one video.



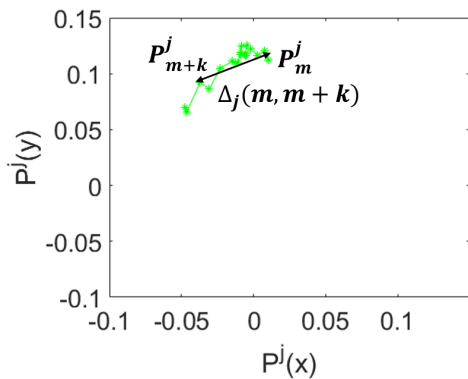
(a) Temporal trajectory of points on whole video  $ROI_j$ , axes are the two first component of PCA. A part of green points (from  $P_m^j$  to  $P_{m+K}^j$ , non-ME frames) and Red points (from  $P_n^j$  to  $P_{n+K}^j$ , ME frames) are zoomed in 3-6d and 3-6b.



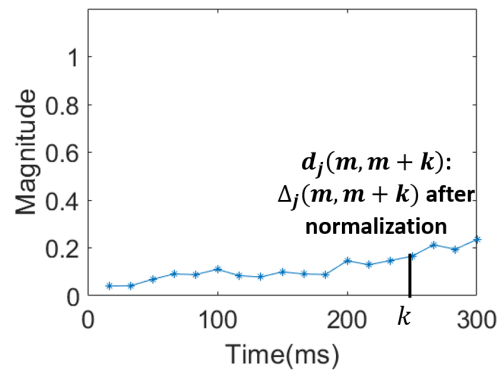
(b)  $K$  points (from  $P_n^j$  to  $P_{n+K}^j$ ) used for pattern computation of ME frame  $n$  on  $ROI_j$ .



(c) Resulting pattern for ME frame:  $LTP_n^j$  (S-pattern).



(d)  $K$  points (from  $P_m^j$  to  $P_{m+K}^j$ ) used for pattern computation of non-ME frame  $m$  on  $ROI_j$ .



(e) Resulting pattern for non-ME frame:  $LTP_m^j$ .

Figure 3-6: Extraction of local temporal pattern. In the sequence  $ROI_j$ , the  $n$ th frame can be represented as point  $P_n^j$  in the PCA projection space. The frame distribution on the PCA projection space of this video sequence is illustrated in 3-6a. (Continued on the next page)

Figure 3-6: The larger the distance between two points is, the more evident the displacement on ROI region between these two frame is. 3-6b shows the point distribution from  $P_n^j$  to  $P_{n+K}^j$  (frames from  $n$  to  $n + K$ , 300ms). Normalized distances between this  $n$ th frame and other  $K$  following frames are calculated as shown in 3-6c. The ensemble of distances forms a curve, which is called the local temporal pattern (LTP). And here is the S-pattern for the ME frames. Meantime, 3-6d shows the point distribution from  $P_m^j$  to  $P_{m+K}^j$  (Non-ME frames from  $m$  to  $m + K$ ). The corresponding LTP pattern is illustrated in 3-6e.



## Distance Computation

The variation of the distances is then studied on the sliding windows of each ROI. The duration taken is 300ms (that is  $K + 1$  frames) to correspond to the average duration of a micro-expression.  $K$  distances between the first frame and the other frames in the interval is calculated. Suppose there are  $K + 1$  frames in the interval, the set of distances in this interval is:

$$\{\Delta_j(n, n + 1), \dots, \Delta_j(n, n + i) \dots, \Delta_j(n, n + K)\}$$

Where  $i \in [1, K]$ ,  $j$  is the ROI index,  $n$  is the index of the first frame in the interval,  $\Delta_j(n, n + i)$  represents the Euclidean distance between the point  $P_{n+i}$  of the  $(i + 1)^{th}$  frame in the interval and the point  $P_n$  of the first frame  $n$  in interval. Therefore, each frame has a dataset which consists of  $K$  distances as listed in Table 3.2. Figure 3-6b and 3-6d respectively show the distance computation for the ME frame  $n$  and non-ME frame  $m$  on the ROI  $j$  (red and a part of green trajectories in Figure 3-6a).

Table 3.2: Distance sets per frame for one ROI video sequence

Frame index	Original distances
1st frame	$\Delta_j(1, 2), \dots, \Delta_j(1, K + 1)$
...	...
$n$ th frame	$\Delta_j(n, n + 1), \dots, \Delta_j(n, n + K)$
...	...
$N$ th frame	$0, \dots, 0$

## Distance Normalization

As the movement magnitudes are not same in different videos, these above distance sets need to be normalized. In average, the distance values reaches the top in 150ms ( $K/2$ ) for the micro-expression. Thus, the normalization is applied depending on the maximum distance in this period for each ROI:

$$\Delta_{j_{max}} = \max_{n=1 \dots N, i=1 \dots K/2} (\Delta_j(n, n + i)) \quad (3.2)$$

Then, the coefficient of normalization (CN) is computed :

$$CN_j = 1/\Delta_{j_{max}} \quad (3.3)$$

And the normalized distance is:

$$d_j(n, i) = \frac{\Delta_j(n, n+i)}{\Delta_{j_{max}}} = \Delta_j(n, n+i) \times CN_j \quad (3.4)$$

The curve formed by the  $K$  normalized distances (blue curve in Figure 3-6c and 3-6e) is the extracted local temporal pattern.

### Configuration of Normalisation Coefficients

The CN value represents the movement magnitude in the entire video sequence for current ROI: while the CN is smaller, the movement is more significant; and vice versa. Figure 3-7 shows the CN value distribution for all ROI sequences in database. Most CN values are smaller than 10.

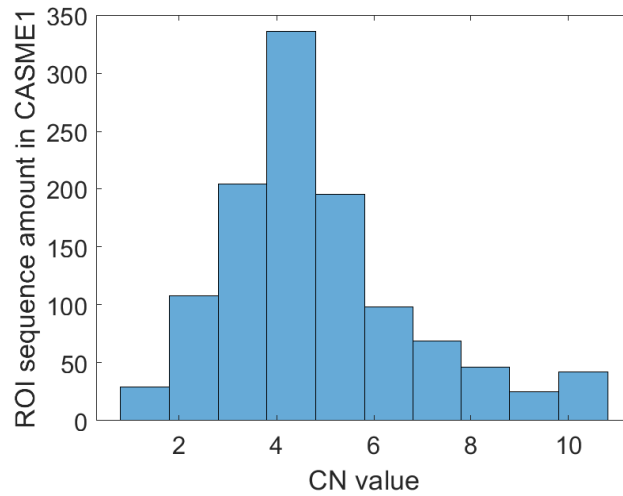


Figure 3-7: Histogram for normalization coefficient (CN) value for all ROI sequences in CASME I. X axis means the CN value, Y axis is the ROI sequence amount for each bin. The average CN value is around 4. Yet, there is a few ROI sequences which have CN values larger than 10, which represents there is almost none evident movement in this video. In the feature construction step, the CN value for this kind of ROI sequences is set to 10.

As the studied samples come from a strictly controlled environment, the head rarely

moved. Therefore, micro-expression motion is normally the most significant movement in ROI. According to the equation 3.3, the bigger the CN is, the more subtle the movements in the ROI sequence are. We consider the ROI sequence with a CN value bigger than 10 as a video clip without any movements. Thus, in our experiment, if the CN value for one certain ROI sequence is larger than 10, then it is set to 10.

### Feature construction

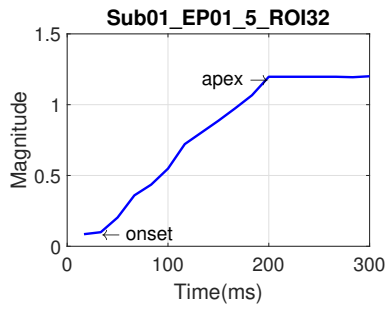
Finally, CN is added into the feature in order to eliminate the movements which are too subtle. Hence, for each frame and each ROI, the final LTP feature is the concatenation of the CN and the  $K$  normalized distances. For instance, the feature composition for the  $n$ th frame of  $j$ th ROI sequence is:

$$CN_j, d_j(n, n+1), \dots, d_j(n, n+K)$$

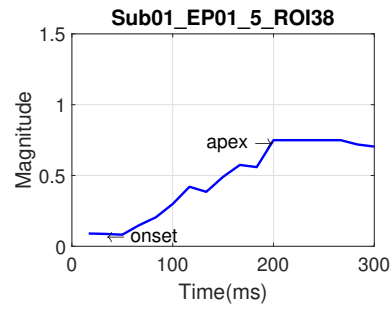
Regardless of the ROI and the subject, each feature is feed into the machine learning (Section 3.3) network separately.

### 3.2.3 S-pattern: a Unique Invariant LTP Pattern for All Micro-Expressions

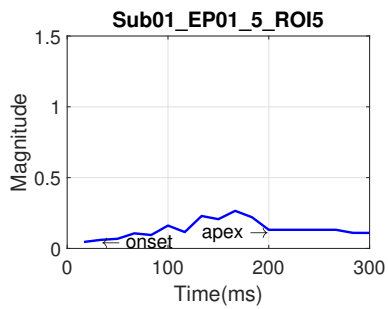
Figure 3-8 shows examples of LTP-patterns in different conditions: different ROI, different subjects, different moment of the video. We can notice that the LTP-patterns are identical when a micro-expression is displayed and forms an S-pattern, regardless of the ROI, the subject and the micro-expression emotion type. Indeed, because the first two components of PCA retain the main variations of the grey level texture in the ROI sequence, the pattern is only influenced by the movement in this local region. Thanks to the invariance of the S-pattern when micro-expression occurs, the local movements can be classified by machine learning into two classes: S-patterns and non S-patterns. A supervised classification SVM is employed (Section 3.3).



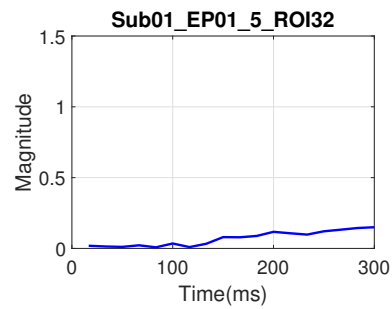
(a)



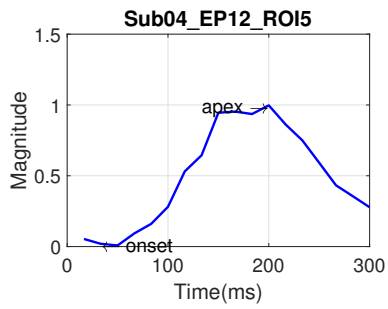
(b)



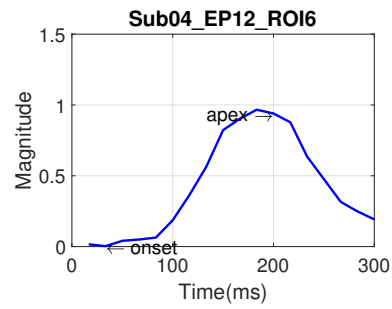
(c)



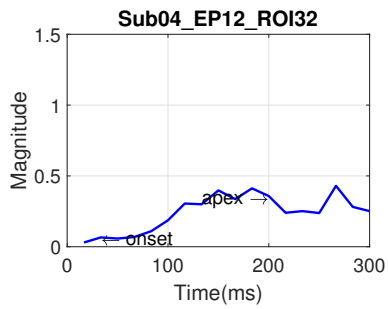
(d)



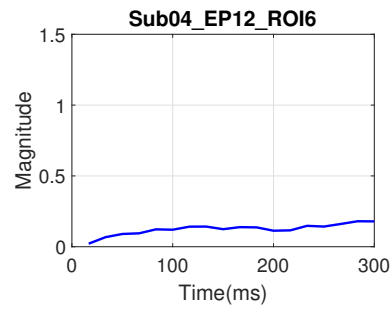
(e)



(f)



(g)



(h)

Figure 3-8: Local temporal patterns (LTPs) of two different videos.(Continued on the next page)

Figure 3-8: Local temporal patterns (LTPs) of two different videos. 3-8a and 3-8b are the LTPs during the micro-expression movement at ROI 32 and ROI 38 (the right and left corners of the mouth) in the video Sub01\_EP01\_5. The emotion of this video is labeled as joy, which is often expressed by mouth corner displacement. 3-8e and 3-8f are the LTPs during the micro-expression movement at ROI 5 and ROI 6 (the inside corners of the right and left eyebrows) in the video Sub04\_EP12. The emotion of this video is labeled as stress, which is often expressed by the movement of the eyebrows. The pattern of curves in these four images are similar even through the ROIs and subjects are different, we call it S-pattern. 3-8c and 3-8g show the LTP of other ROIs at the same time as 3-8a/ 3-8b and 3-8e/ 3-8f respectively. The pattern is different from S-pattern because the micro-expression does not occur on these regions. 3-8d and 3-8h illustrate the LTPs in the same ROI as 3-8a and 3-8f respectively, but at a different moment in the video. These patterns differ from S-pattern since the micro-expression does not occur at this moment. (video samples in CASME I)

### 3.3 Micro-Expression Spotting: Classification and Fusion

The micro-expression spotting is processed by two steps: a local LTP classification and a spatial and temporal fusion (see Figure 3-1). Concerning the LTP classification, the LTPs are firstly selected for efficient machine learning training (subsection 3.3.1). Then, LTPs are classified as S-pattern or non S-pattern at the test stage (subsection 3.3.2) . Finally, a fusion analysis from local to global is performed (subsection 3.3.3) to get one global result for each frame. It aims at both eliminating global head movement and also merging the local positive results from different ROIs that belongs to the same global micro-expression.

#### 3.3.1 LTP Selection for Training

For training stage, LTPs should be separated into 2 groups: S-pattern (micro-expression) and non-S-pattern (other facial movement). In the public databases, the micro-expression sequences are annotated with the onset and AU information. Thus, we need to pre-process the labels of the databases to get the ground truth for our training. As shown in Figure 3-9, LTPs pass through a pre-processing with 3 steps: a temporal annotation, a local selection (AU selection per ROI) and an LTP shape related selection.

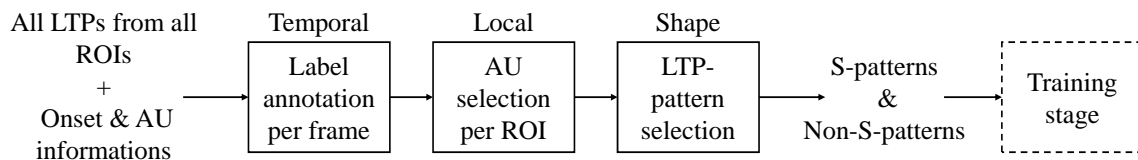


Figure 3-9: LTP selection for training stage. All the LTPs are classified into 2 classes: S-patterns and non-S-patterns. LTPs pass through 3 steps for the annotation: label annotation per frame, AU selection per ROI and LTP pattern selection. The annotated S-patterns and non-S-patterns are then fed to the training stage of the SVM classifier.

#### Label annotation per frame

To label the database, the  $K/3$  frames before micro-expression onset are found to retain the best pattern, where  $K + 1$  is the length of interval as mentioned in the above section and  $K/3$  is an empirical value. Hence, the frames are annotated as shown in Figure 3-10.

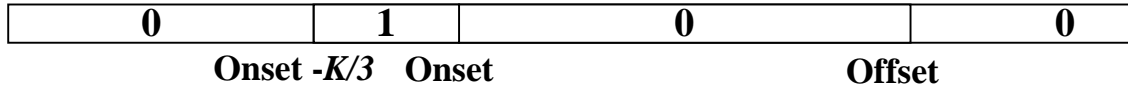


Figure 3-10: Label annotation per frame. The rectangle represents an entire video, and the interval from onset to offset is the ME sequence. The S-pattern is expressed at the beginning of the onset. Hence, frames with S-pattern (in the range  $[\text{onset}-K/3, \text{onset}]$ ) are labeled with label 1 (S-pattern) and the other frames are labeled with 0 (non S-pattern).

In addition, the sample distribution of S-patterns and other LTPs (non-S-patterns) is not well balanced, there are more than 10 times of other LTPs than the amount of S-pattern. Thus, the other LTPs are sub-sampled by a rate of 1/8 for model training.

### AU selection per ROI

The LTP pattern is extracted from each ROI per frame. For the frames which is labeled as micro-expression, not all the patterns from these frames fit the S-pattern since micro-expression is a local movement. Hence, the ROI index for training is firstly selected based on the AU annotation info for each video, as illustrated in Table 3.3. For instance, if the AU annotation is AU4, as the index 4 is smaller than 6, the selected ROIs for training are ROI 1,4,5,6,7,10. This selection is a weak selection, for some ROI may not have any movement even if they are located on the corresponding upper or lower part of the face.

Table 3.3: ROI selection for training stage of machine learning. Since micro-expression is a local movement, ROIs which have annotated AU are chosen to represent the micro-expression movement.

AU condition	ROI index	Facial region
All given AU index <6	1,4,5,6,7,10	Eyebrows
All given AU index >9	32,35,38,41	Mouth contour
Otherwise	all chosen ROIs	Entire face

### LTP pattern selection

Hence, after the AU selection, a further LTP pattern selection process is performed. Three criteria are enforced, as shown in Figure 3-11. All S-patterns which do not fit at least one

of the requirements would be deleted from S-pattern training dataset. First of all, since the micro-expression is the most significant movement on the chosen ROI after the PCA process, the distance value  $d_j$  is a criteria for S-pattern selection. If the maximum distance value is smaller than a threshold distance  $T_{dist}$  (found empirically), then the S-pattern is removed. Secondly, depending on formula 3.3, the ROIs which generate micro-expression should have lower CN values than other ROIs of the same face at the same time. The average value of CNs ( $\bar{CN}$ ) for ROIs is set as threshold to select proper patterns. Finally, as introduced in Figure 1-2, the S-pattern has a certain curve shape. The curve slope of onset frame should be positive in the first 150ms, then the slope becomes zero or negative in the following 150ms.

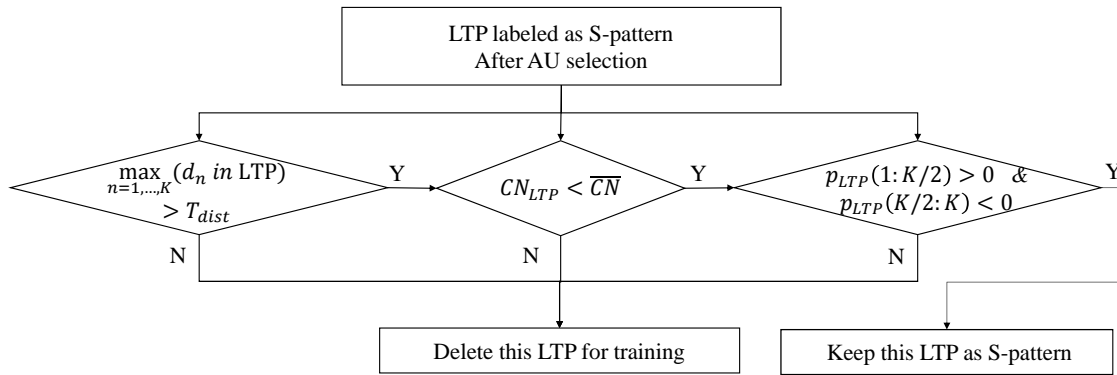


Figure 3-11: LTP pattern selection of training stage for local classification. LTPs labeled as S-pattern pass through this process to conserve reliable S-patterns. The selection criteria include distance value  $d$  in LTP, normalization coefficient ( $CN$ ) and curve slope( $p_{LTP}$ ).

**The LTP selection step construct the training dataset: label annotations identifies the S-patterns and non-S-patterns, than the AU selection and LTP selection help to conserve reliable S-patterns for training the classifier.**

### 3.3.2 LTP Classification - SVM

Once the feature annotation (S-pattern and non S-pattern) is performed, a supervised classification SVM is employed. The results of local classification are generated by LOSubOCV (Leave-One-Subject-Out Cross Validation). ROIs with S-pattern are recognized, indicating that a movement similar to micro-expression occurs in this region.



### 3.3.3 Spatial and Temporal Fusion

The local information per frame is obtained by the LTP classification. Yet, the spotting result should be a global information. Thus, we perform a spatial and temporal fusion. The process includes three steps: local qualification, spatial fusion and merge process. This fusion process is illustrated in Figure 3-12.

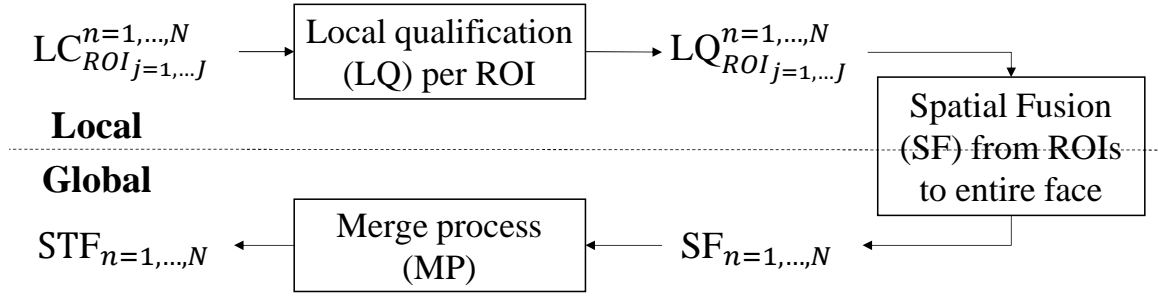


Figure 3-12: Spatial and temporal fusion. The predicted labels from local classification for all ROIs are represented as  $LC_{ROI_{j=1,...,J}}^{n=1,...,N}$ , where  $N$  is the number of frames in the whole video,  $J$  is the chosen ROIs amount. Passing through the local qualification (LQ) per ROI sequence, the spotting intervals which are too short or too long are deleted. Then for each frame  $n$ , the local spotting results  $LQ_{ROI_{j=1,...,J}}^n$  are integrated into a single  $SF_n$ , which represents the spotting result for the entire frame. A merge process is applied on  $SF_{n=1,...,N}$  to form a consecutive ME movement. Thus, we get the final result  $STF_{n=1,...,N}$  for one video sequence.

#### Local Qualification

First of all, two thresholds  $T_{CN}$  and  $T_{dist}$  are set to enhance the ability of differentiating S-pattern from other LTPs. If the CN value for the recognized S-pattern is bigger than  $T_{CN}$  or the maximal distance value ( $max(d_j)$ ) of this S-pattern is smaller than  $T_{dist}$ , the movement related to this S-pattern will not be considered as ME. Furthermore, a temporal selection is performed. The frame number with the label 1 is limited locally in an interval of length  $K$ . In fact, the duration of micro-expression is normally around  $K$  frames, and the optimal condition is to detect all  $K/3$  frames before the micro-expression onset. Having less than  $K/9$  or more than  $K/2$  patterns detected is not considered as micro-expression. Therefore, the number of frames detected as label 1 is limited to  $K/9 - K/2$ . The detected frames are searched by a sliding non-overlap window, and only the frames that fit the criteria can

be conserved. For the purpose of reproducibility of the algorithm, the flow chart of this temporal selection process is illustrated in Figure 3-13.

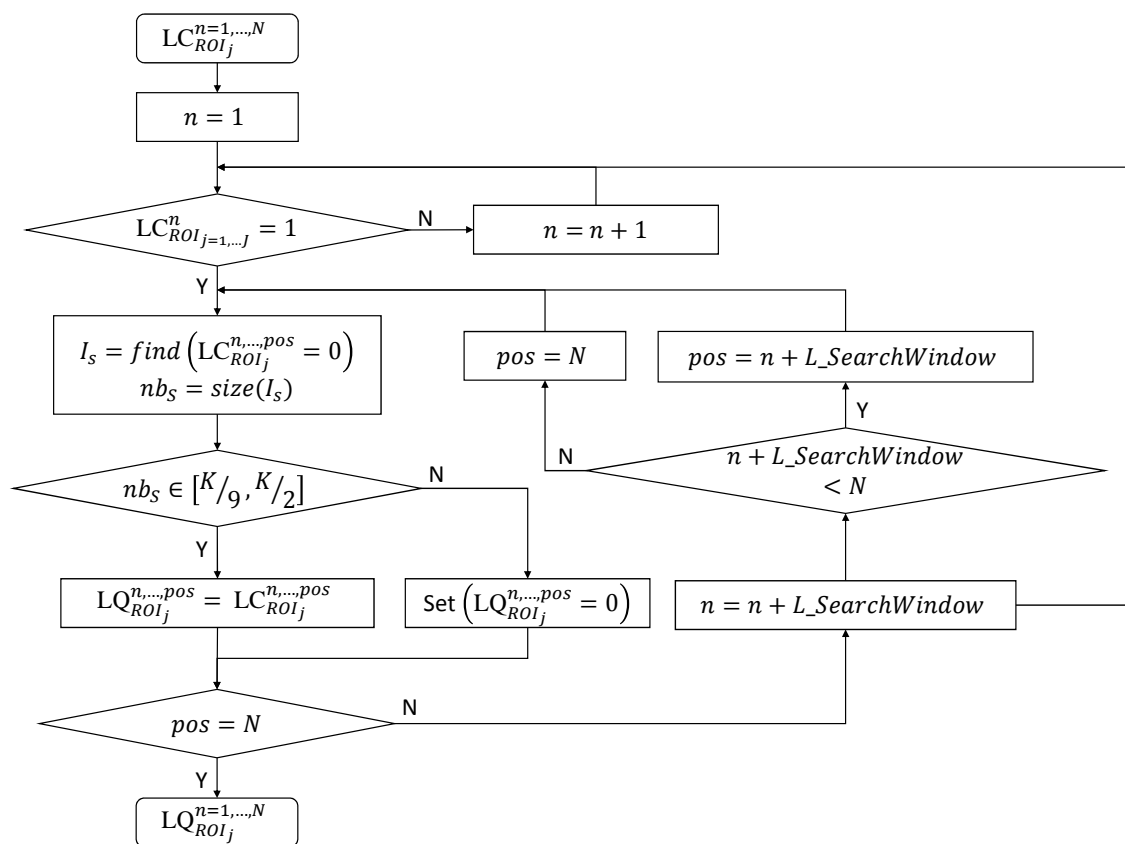


Figure 3-13: Flow chart of temporal selection process in local qualification. The  $j_{th}$  ROI sequence passes through the distance and normalization coefficient threshold selection, then predicted label of this sequence  $LC_{ROI_j}^{n=1,...,N}$  enter the process as input.  $LQ_{ROI_j}^{n=1,...,N}$  is the output of the process, i.e. the result after local qualification (LQ).

## Spatial Fusion

Secondly, a spatial fusion is performed. The recognized S-patterns from local regions are integrated into a global spotting result. Indeed, micro-expressions are local movements. This spatial fusion process aims at eliminating head movement. The rule is the following one: if there are more than  $J/2$  ROIs of entire face or more than one nose region that have been detected with some movement, this motion is then considered as a global head movement. What's more, the eye blinking leads to all the muscles around the eye. Thus, if all the ROIs of eyebrows detect movement, the micro-expression spotting system supposes

there is an eye blinking, and treat the current frame as non-micro-expression. For the purpose of reproducibility of the algorithm, the entire process is shown in Figure 3-14.

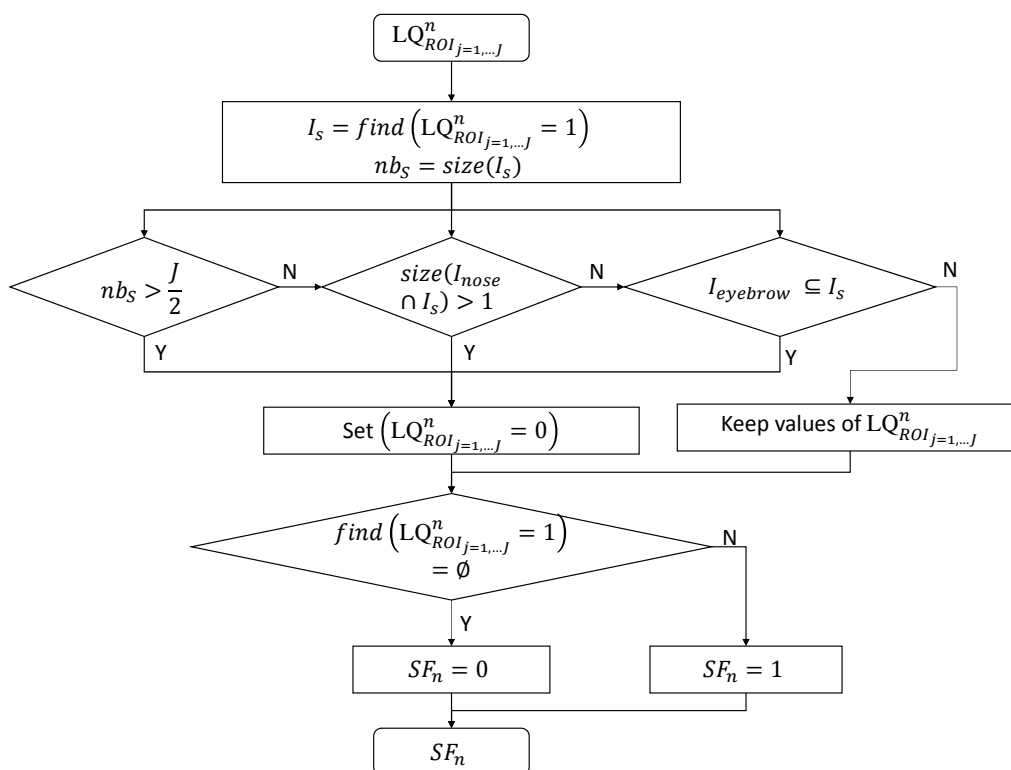
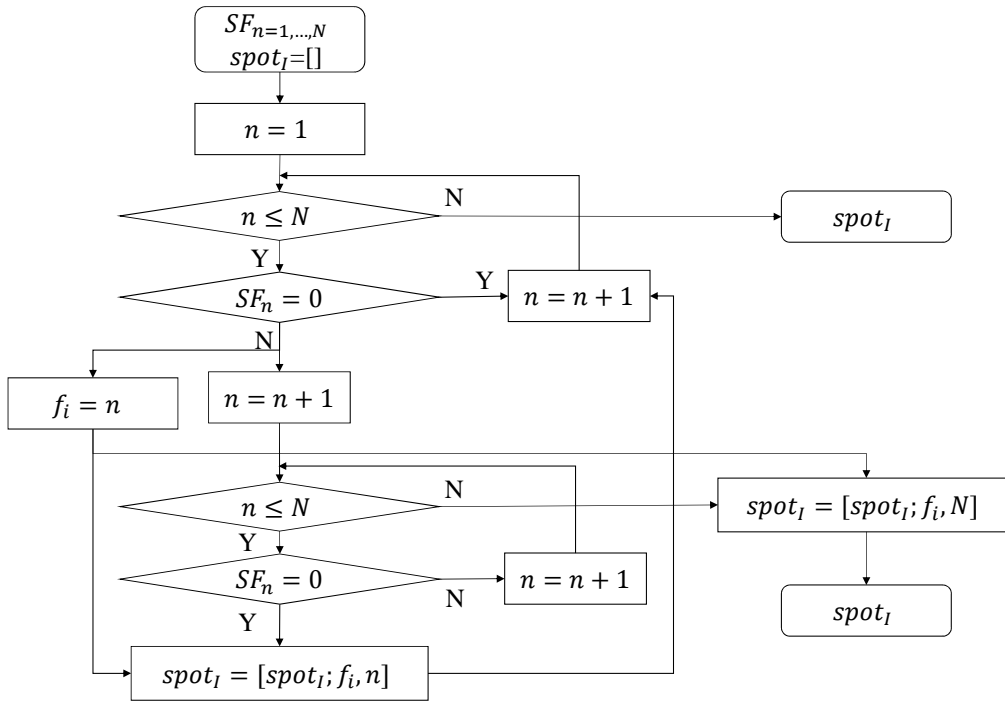


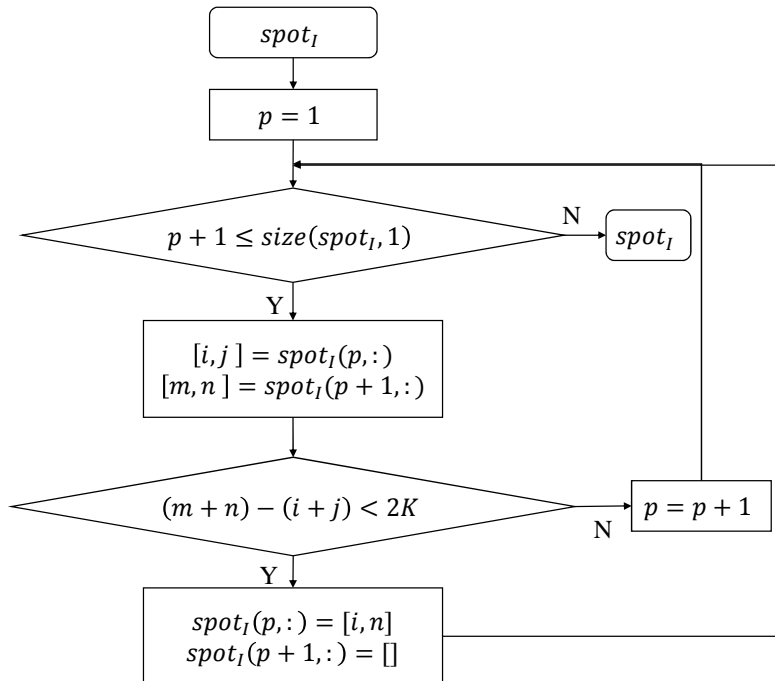
Figure 3-14: Flow chart of spatial fusion.  $I_{nose}$  means the all the ROI index on nose region, and  $I_{eyebrow}$  means all the ROI index on eyebrows. For  $n_{th}$  frame in video sample, predicted label of all  $J$  ROIs  $LQ_{ROI_{j=1,...,J}}^n$  enter the spatial fusion process. The output  $SF_n$  is a predicted label for this frame, and it represents that whether there is micro-expression on this entire frame or not.

### Merge Process

Thirdly, the S-pattern is until now recognized per frame. The result is not a consecutive interval. To get more robust results, the nearby frames classified as ME are merged with proper criteria. For two adjacent spotted intervals with their frames indexes:  $[f_i, f_j]$  and  $[f_m, f_n]$ , if the distance in terms of frame number between their midpoints is smaller than  $K$ , i.e.  $(m + n) - (i + j) < 2K$ , then these two intervals are merged to a longer interval, in which all the frames are predicted as ME. The flow chart is shown in Figure 3-15 for the reproducibility of the algorithm.



(a) The frame indexes (i.e. first and last frame) of spotted intervals are noted in  $spot_I$ .



(b) The noted frame indexes in  $spot_I$  are merged in 3-15b. After the merge process,  $spot_I$  contains the final spotted intervals.

Figure 3-15: Flow chart of merge process.

Since the S-pattern is unique for all kinds of micro-expressions, we can use machine learning method to classify local movements with S-patterns or non-S-patterns. The LTP selection constructs the training dataset, then the SVM classifier is used to spot S-patterns on local regions. Finally, the spotting result is obtained by this spatial and temporal fusion. The local qualification and the spatial fusion reduce the spotted movements which are not micro-expressions, then the merge process enhance the robustness of our method.

### 3.4 Spotting Micro-Expression in Long Videos

The previous work focus on the micro-expression spotting in short videos (less than 2s). However, in the real life, the videos for micro-expression analysis are much longer. Hence, it is necessary to develop and test our method in the long video situation. There are two spontaneous micro-expression databases which contain long videos: SAMM [19] and CAS(ME)<sup>2</sup> [104]. All the experiments of the following research are performed on these two databases.

This section introduces the modification of our method for the applications in long Videos, including two steps in pipeline figure: pre-processing (subsection 3.4.1) and ME spotting (subsection 3.4.1).

#### 3.4.1 Pre-Processing in Long Videos

As micro-expression is a local facial movement, we analyze micro-expression only on a selection of regions of interest (ROIs). This subsection introduces the ROI sequence extraction in long videos.

Since the samples in SAMM and CAS(ME)<sup>2</sup> were recorded in the strictly controlled laboratory environment, the subjects barely moved in one second. As the average duration of micro-expression is around 300ms, the long video in these two databases are processed by a temporal sliding window  $W_{video}$  whose length is 1s. The overlap is set to 300ms to avoid missing any possible micro-expression movements. Thus, the video is divided into an ensemble of small sequences  $[I_1, I_2, \dots, I_M]$  by sliding temporal window as shown in Figure 3-16, where  $M$  is the total number of small sequences.

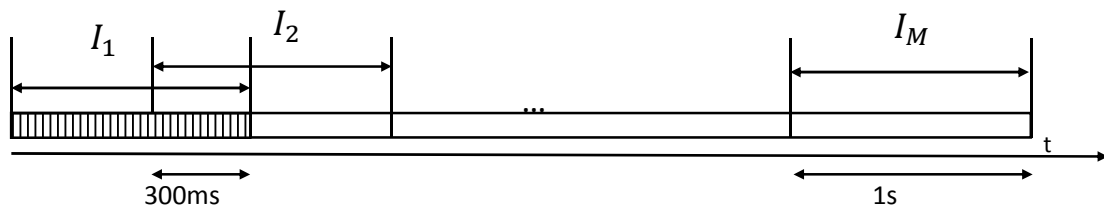


Figure 3-16: The long video is divided into several short sequences ( $I_1, \dots, I_m, \dots, I_M$ ) by a sliding window (1s).

For each  $I_m$  ( $m \leq M$ ) video sequence, the basic pre-process is the same as presented in section 3.1. First of all, 84 facial landmarks are tracked in the video sequence by utilizing the Genfacetracker (©Dynamixyz) [2]. Genfacetracker is utilized instead of Intraface, because it detects the landmarks more precisely than Intraface in long videos. Then 12 ROIs in size of  $a = (1/5) \times L$  are chosen as the same process presented in sub-section 3.1. Figure 3-17 illustrates the two facial images after pre-process in SAMM and CAS(ME)<sup>2</sup>. In addition, the positions of ROI sequence  $ROI_j^m$  ( $j \leq 12$ ) in  $I_m$  short video are determined by the detected landmarks of the first frame in  $I_m$ .

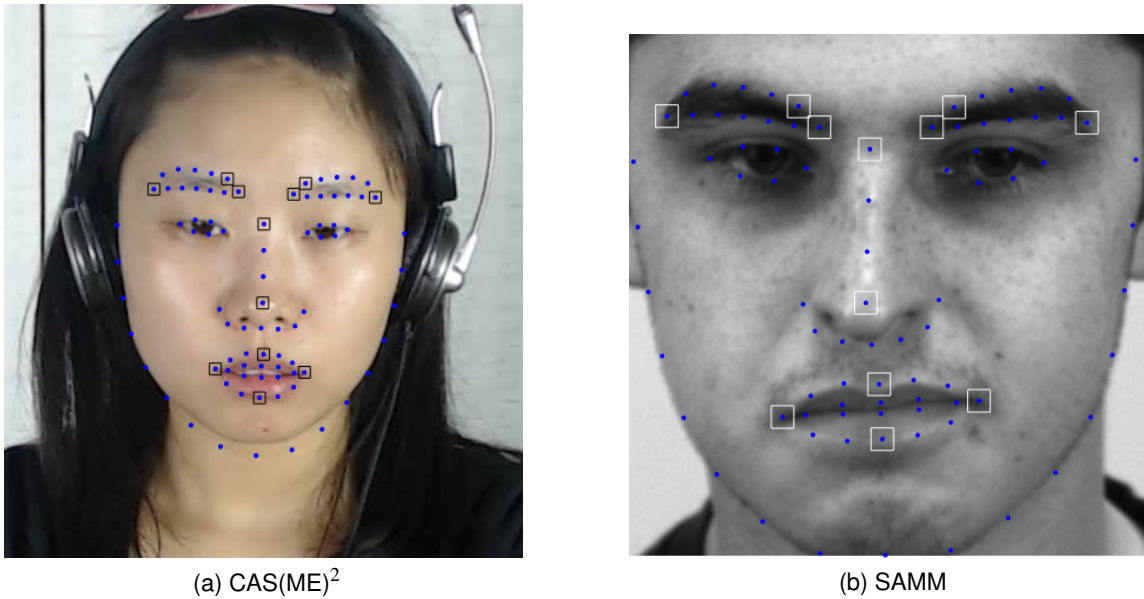


Figure 3-17: Facial landmarks tracking and ROI selection. On the right: an example from SAMM; on the left: an example from CAS(ME)<sup>2</sup>

### 3.4.2 Obtaining Spotting Result in a Long Video

After the pre-processing, the following process for micro-expression spotting is the same as the process for short videos (presented in section 3.2 and 3.3). The subscript of formulas in LTP feature extraction is changed, because there multiple short sequences in one long video. The detailed process is introduced in AppendixC. LTP for each  $ROI_j^m$  is classified as S-pattern or non-S-pattern. After the spatial and temporal fusion, we obtain the spotting result for the short video  $I_m$ .

Results from all short videos are then merged to get the final spotting for the entire long video. Concerning the overlap region, suppose the spotting result of preceding sequence for this region is  $S_{i=1,\dots,K}^m$ , and that of following sequence is  $S_{i=1,\dots,K}^{m+1}$ , then for frame  $i$ , the spotting result is positive as long as  $S_i^m > 0$  or  $S_i^{m+1} > 0$ .

**In this section, we apply our proposed method in long videos to spot micro-expressions. The long video is firstly divided into short sequences with an overlap, then the our method is performed to spot micro-expressions. The final spotting result for the long video is obtained by merging the results of short sequences.**



## 3.5 Conclusion

Our proposed method spots micro-expressions using a local temporal pattern of facial movement, which is the same pattern for all the ROIs and all the micro-expression types. Micro-expressions can be distinguished from other movements by this pattern. A supervised learning algorithm is utilized to achieve this goal. In addition, this pattern allows us to identify the spatial location where the micro-expression occurs. Moreover, PCA is performed to facilitate the classification through the SVM by reducing the data dimension and removing the noise. Meantime, the spatial and temporal fusion from local to global enhances the ability of differentiating ME from other facial movements.

Yet, the performance of our method may be restricted if the dataset is not big enough. Unfortunately, available databases give very few micro-expression samples, as mentioned in Chapter 2. For instance, only 78 LTPs are qualified as S-patterns by the basic LTP selection and then used for micro-expression spotting in CASME I-section A. Hence, in order to efficiently increase the size of S-pattern dataset, we propose to filter LTPs and then synthesize more S-patterns by Hammerstein model, that will be added to the training stage.

# Chapter 4

## Data augmentation using Hammerstein

### Model

As introduced in previous chapter, our method is limited by the amount of micro-expression (ME) samples. There are not many labeled ME databases, hence not many labeled ME frames. In addition, the amount of non-ME frames is much larger than that of ME frames, i.e. the size of S-pattern dataset is not big enough. By synthesizing the features of micro-expression, the training data volume can be extended. Since micro-expressions are brief and local facial expressions, the deformation intensity is not evident while the duration is very short. Thus, generating directly facial images may import the generation error into the texture feature extraction, and the brief temporal variation between frames is difficult to simulate. As a consequence, we propose to synthesize micro-expression features for the data augmentation.

In this chapter, to improve the performance of our method, we propose to increase the size of S-pattern dataset for the training stage of the local classification. To that purpose, we use Hammerstein model (HM), which is well known to simulate the dynamic of muscle movement. Figure 4-1 illustrates the modification in the whole process: LTP pattern selection is replaced by LTP filtering and S-pattern synthesizing. More precisely, label annotation and AU selection from LTP selection in Figure 3-9 are kept as preceding steps. Then, we replace the third step, 'LTP pattern selection' by 'LTP filtering and S-pattern synthesizing by Hammerstein model'. The scheme of this novel sub-process is illustrated in

Figure 4-2. The S-patterns coming from label annotation and AU selection ( $S\text{-patterns}_O$ ) are firstly modelled by the system identification of Hammerstein model. They are then filtered, and the remaining selected  $S\text{-patterns}_{OF}$  serve as seeds for synthesizing more S-patterns ( $S\text{-patterns}_{ST}$ ).

After a brief introduction of Hammerstein model (subsection 4.1), the model is applied to S-pattern (subsection 4.2). Afterwards, LTP filtering and S-pattern synthesizing are presented respectively in subsection 4.3 and 4.4.

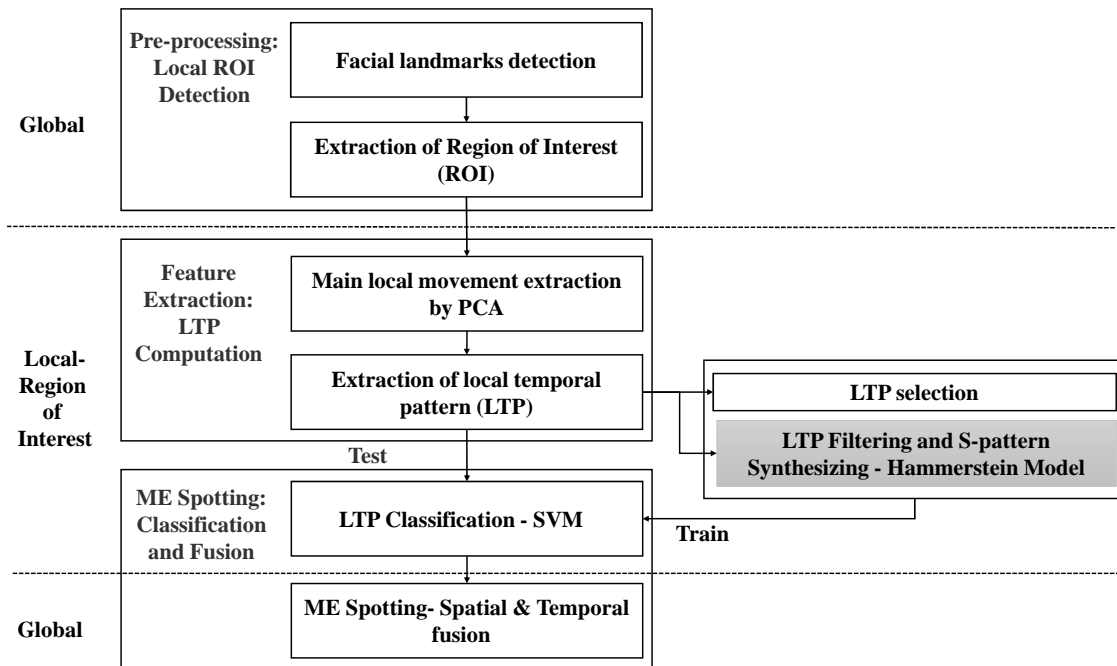


Figure 4-1: Overview of our method combining Hammerstein model. Our proposed method has been presented in Figure 3-1. The grey block replaces the LTP selection by using Hammerstein Model. More reliable S-patterns (LTP patterns specific to ME movements) are produced by this model. LTPs including S-patterns (both real and synthesized) are then used as the training samples to build the machine learning model (SVM) for classification.

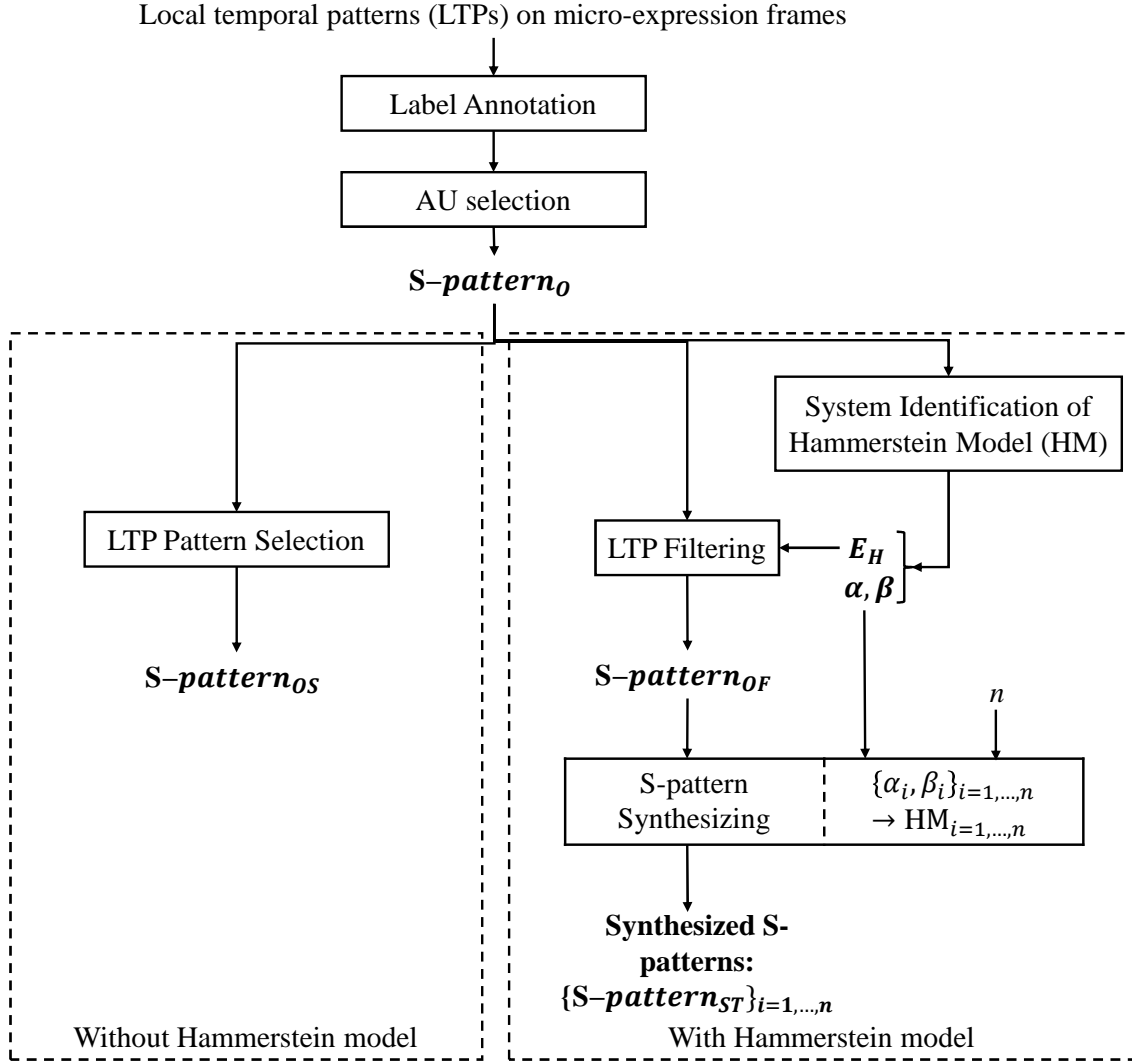


Figure 4-2: LTP filtering and S-pattern synthesizing by Hammerstein model (HM) during the training stage. In the right block, the original S-pattern ( $S\text{-pattern}_O$ , after label annotation and AU selection in Figure 3-9) passes through the system identification of Hammerstein model. The parameter set  $(\alpha, \beta, E_H)$  corresponding to this S-pattern is estimated. S-patterns are selected by LTP filtering process according to the estimation error  $E_H$ . The selected patterns ( $S\text{-pattern}_{OF}$ ) are used to generate  $n$  synthesized S-patterns ( $S\text{-pattern}_{ST}$ ). For comparison, the left block shows our method without Hammerstein model, i.e. the result after LTP pattern selection:  $S\text{-pattern}_{OS}$ .

The below abbreviation are frequently used:

$S\text{-pattern}_O$ : original S-pattern after label annotation and AU selection of Figure 3-9.

$(\alpha, \beta)$ : parameters in the linear module of Hammerstein model.

$E_H$ : Estimation error of Hammerstein model

$S\text{-pattern}_{OF}$ : Conserved  $S\text{-pattern}_O$  after LTP filtering.

$S\text{-pattern}_{ST}$ : Synthesized S-pattern by Hammerstein model.

$S\text{-pattern}_{OS}$ : Conserved  $S\text{-pattern}_O$  after LTP pattern selection of Figure 3-9.

## 4.1 Hammerstein Model

In order to use Hammerstein model for data volume augmentation, first of all, the theory and the mathematical formulas of this model need to be studied. Thus, in this section, the Hammerstein model is briefly introduced.

Hammerstein model [11] is a popular black box model in biomedical engineering. It is a traditional simulation model, which is based on the solid mathematical formulas and has physic explications. The model has been successfully used in [18, 109] for the modeling of isometric muscle dynamics. Figure 4-3 show a Hammerstein model which can represent a discrete-time electrically stimulated muscle model.

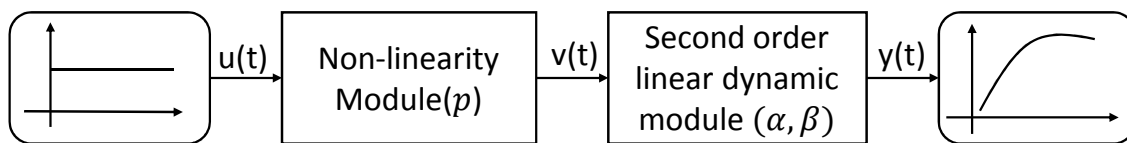


Figure 4-3: Hammerstein model structure. Hammerstein model represents well the muscle movement. S-patterns are caused by the facial muscle movement and the variation curve is similar to the local muscle movement. Thus, S-patterns can be well synthesized by Hammerstein model. The model is a concatenation of two modules: a static non-linearity module (that manipulates the magnitude) and a second-order dynamics linear module (that simulates the movement pattern).

The above mentioned Hammerstein model contains two modules in series: a non-linearity module preceding a second-order dynamics linear module. The input nonlinear module represents the magnitude of the deformation and the stimulated muscle dynamics are determined by linear module. The complexity of the model is the amount of parameters. The more complex the system is, the more precise the estimated behavior is. However, it is difficult to identify all parameters in the model. According to [11], the dynamic properties can be described accurately with a second order linear module. The performance is largely better than that of 1-order, but just slight worse than the model with a third-order linear module. The second-order linear module is more competitive in terms of the number of parameters for Hammerstein model. This is why we choose it for our method. The transfer

function of the linear module in our method is then:

$$\frac{Y(s)}{V(s)} = \frac{s}{1 + \alpha s + \beta s^2} \quad (4.1)$$

where  $\alpha$  and  $\beta$  are module parameters.  $Y(s)$  and  $V(s)$  are the Laplace transform of  $v(t)$  and  $y(t)$ , which are the input and output of the linear module (see Figure 4-3).

In conclusion, the Hammerstein model is a well-developed traditional simulation model. For our purpose, the Hammerstein model can be characterized by the error and the parameters of two sub modules:

- $p$  for the non-linear module;
- $(\alpha, \beta)$  for the linear module
- the global estimation error  $E_H$ .

The parameters  $(\alpha, \beta)$  in the linear module can help to manipulate the dynamics of facial movement.

In conclusion, the data augmentation by Hammerstein model utilizes the traditional simulation model, with solid physic explications. Yet, concerning the the emerging generation method: GAN [33] (Generative Adversarial Network), the interpretability of this method is weak. The performance is depending on the selection of the training set and the parameter manipulation. The model construction is empirical and it is difficult to explain the result. However, concerning the applications of medical care or national security, the reliability of the system should be ensured. Since the mathematical theory of deep learning is still on exploring, we choose to use well-developed traditional simulation model: Hammerstein model to synthesize S-patterns. More experimental analysis for the comparison between Hammerstein model and GAN is presented in subsection 5.4.2.

## 4.2 Applying Hammerstein Model to S-Pattern

In this section, we analyze the values and the impacts of the parameters of Hammerstein model to build model for S-patter. The analysis allows us to filter wrongly-labeled S-patterns (Section 4.3) and then synthesize more S-patterns (Section 4.4).

Firstly, the system identification is introduced to identify the parameters in the non-linear module and the linear module. Then, we present the estimator of the non-linear module and indicate that parameters  $p$  in this module has little influence on dynamic property of micro-expression. Finally, in the last two sub-sections, the distribution of  $(\alpha, \beta)$  of the linear module and the estimation error  $E_H$  are investigated and associated the curve shape of S-pattern.

### 4.2.1 System Identification

The system identification enables to identify the parameters of two modules in Hammerstein model based on measured input-output data. The process is illustrated in 4-4. In our case, the input  $u(t)$  in formula 4.1 is a constant command, and the output  $y(t)$  is an original S-pattern ( $S\text{-pattern}_O$ ). Once the input and output are fixed for the model, the system parameters of the two sub-modules, i.e  $(p, \alpha, \beta)$  can be identified. With the estimated Hammerstein model and a constant command, we can synthesize a virtual S-pattern ( $S\text{-pattern}_{ST}$ ) which is similar to  $S\text{-pattern}_O$ . Figure 4-5 shows an example of the original S-pattern ( $S\text{-pattern}_O$ ) and synthesized S-pattern ( $S\text{-pattern}_{ST}$ ) by Hammerstein model. The detailed process is presented in following paragraphs.

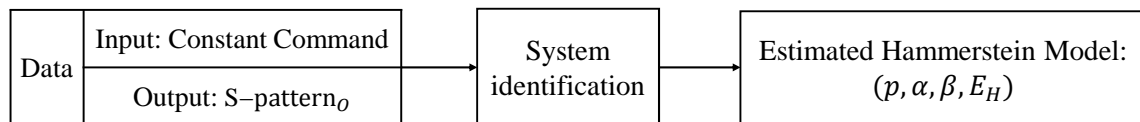


Figure 4-4: The basic system identification process for Hammerstein model. Depending on the data which is constructed by constant command and chosen  $S\text{-pattern}_O$ , the corresponding Hammerstein model can be estimated by system identification. In other words, the parameters of the non-linearity module ( $p$ ), of the linear module  $(\alpha, \beta)$  and the system estimation error ( $E_H$ ) are determined.

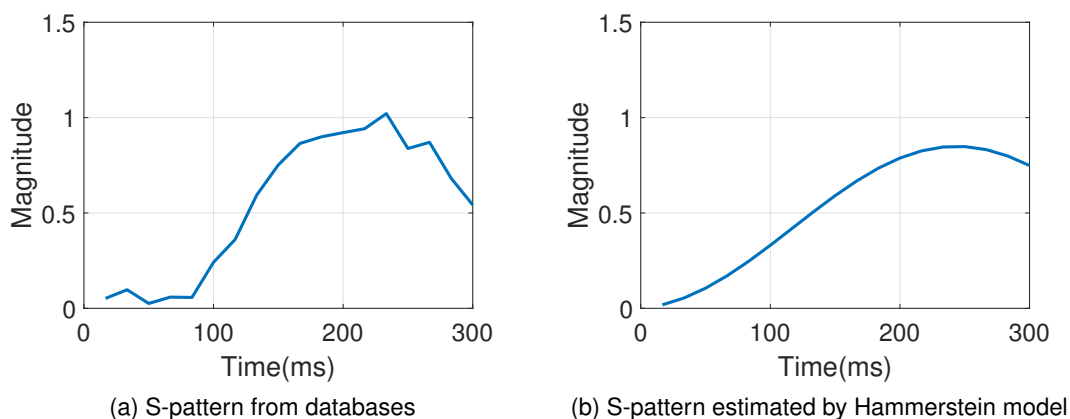


Figure 4-5: One Example of real S-pattern ( $S\text{-pattern}_O$ ) and the corresponding synthesized S-pattern ( $S\text{-pattern}_{ST}$ ) by estimated by Hammerstein model.

**Data Construction** A time domain data object is constructed for the system identification, where  $u$  and  $y$  are time domain input and output respectively (see Figure 4-3),  $Ts$  is the sample times for the experiment.

- $u(t)$ : Since the micro-expression is a local brief facial movement, the motion intensity is low. Hence, the forces to trigger the displacement from different ROIs are small and the difference between these forces can be ignored. As a consequence, the input of Hammerstein model for S-patterns from all ROIs is set to the same constant command.
- $y(t)$ : S-pattern is considered as the output of the system as its curve shape is our target for synthesizing.
- $Ts$ : The starting time is set to zero considering that the spontaneous micro-expression occurs unconsciously and very rapidly. The sample time for experimental data is set as the reciprocal of FPS of the video sequence, i.e.  $Ts = 1/FPS$ .

After the data construction, the data object ( $y(t), u(t), Ts$ ) is imported into the system identification.

**Model Design and estimation** During this step, we design the basic setting of Hammerstein model, i.e. the type of the non-linear module and order of the linear module. The non-



linearity module for input channels is set as the pwlinear (piecewise-linear) estimator with 10 breakpoints (default). A focus on non-linear module types is given in subsection 4.2.2. Meanwhile, the order of the linear module is set to 2 as introduced in Subsection 4.1. The parameters of the linear module are analyzed in subsection 4.2.3.

**Estimated Hammerstein Model** The structure of the estimated Hammerstein model contains all the parameters of the two modules and the performance report:  $p, \alpha, \beta, E_H$ . According to equation 4.1,  $(\alpha, \beta)$  can determine the linear module and hence influence the entire Hammerstein model. Since the S-pattern is a 1-D curve, MSE (mean squared error,  $E_H$ ) is sufficient for the estimation evaluation.

## 4.2.2 Parameter for the Non-Linear Module

**The non-linear module controls the magnitude of the output signal. As the S-patterns are obtained after a normalization process, the variations of  $p$  for different S-patterns are subtle, i.e.  $p$  has little influence on S-pattern synthesizing.**

Yet, to construct the Hammerstein model, the configurations of the non-linear module need to be clarified. A well-chosen estimator can help to improve the efficiency of the system identification process and reduce the estimation error.

Concerning the non-linear module, there are several estimation representations [3], including pwlinear, sigmoidnet, poly1d, saturation, waveNET and deadZone. In our application case, there are only 18 distance values for S-pattern extracted from ME video sequence of 30FPS. WaveNet is not suitable for our case, since there are too few data samples for initializing the WAVENET object. Thus, except waveNet, all the other estimators have been tested by system identification with an original S-pattern ( $S\text{-pattern}_O$ ) as input. After the system identification, an Hammerstein model is estimated. Meantime, an synthesized S-pattern ( $S\text{-pattern}_{ST}$ ) is generated by this Hammerstein model. The fit rate ( $R^2$ , pronounced r-square) of  $S\text{-pattern}_{ST}$  to the  $S\text{-pattern}_O$ , the result of loss function and the computation time are used to evaluate these non-linear estimators. Table 4.1 shows the evaluation result. Pwlinear and sigmoidnet are both competitive with a high fit rate and short computation time. To choose between these two, we use more metrics: LossFcn (the result of loss func-

tion) and computation time to evaluate them. 78 S-patterns<sup>1</sup> (S-pattern<sub>OS</sub>) are used as an entire input to identify the Hammerstein model with these two estimators. The results of loss functions are 0.1131 and 0.1175 respectively. As a result, pwlinear [4] is chosen as the estimator for the non-linearity module.

Table 4.1: Estimation performance of different non-linearity modules in Hammerstein model. Fit rate is the fit rate of generated pattern to original pattern; LossFcn means the result of loss function. All the estimators are performed with default parameters.

Non-linear Module	Fit rate	LossFcn	Time
Pwlinear	<b>89.84%</b>	<b>0.001747</b>	2.335s
Sigmoidnet	<b>89.84%</b>	<b>0.001747</b>	<b>2.246s</b>
Poly1d	89.56%	0.001773	0.248s
Saturation	89.84%	0.00175	2.452s
DeadZone	-82.53%	0.6412	1.821s

### 4.2.3 Relationship between the Linear Module and S-patterns

**As the dynamic property of the output signal is controlled by the linear module, we emphasize on  $(\alpha, \beta)$  and analyze the influence of their variation on the curve shape of the S-pattern.**

The relationship between parameters of Hammerstein model<sup>1</sup> and original S-pattern is investigated on CASME I-section A. 78 Hammerstein models with their corresponding  $(p, \alpha, \beta)$  have been estimated by system identification for this analysis.

As illustrated in Figure 4-6, the distribution of  $(\alpha, \beta)$  is related to the curve shape of the S-patterns. The  $(\alpha, \beta)$  points, which are on the upper-left side of the figure, are linked to the most representative shapes of S-pattern. In addition,  $E_H$  is the mean squared error between the S-patterns after LTP pattern selection (S-pattern<sub>OS</sub>) and the synthesized S-pattern (S-pattern<sub>ST</sub>) estimated by the Hammerstein model. The estimation errors  $E_H$  of the points on the upper-left side of the figure are acceptable. Hence, we would be able to both filter real LTPs depending on  $E_H$  values and also synthesize the virtual S-patterns based on the upper-left zone of the distribution figure.

<sup>1</sup>These patterns are the S-patterns of CASME I database that have been conserved after the LTP pattern selection process (S-patterns<sub>OS</sub>) (Subsection 3.3.1)

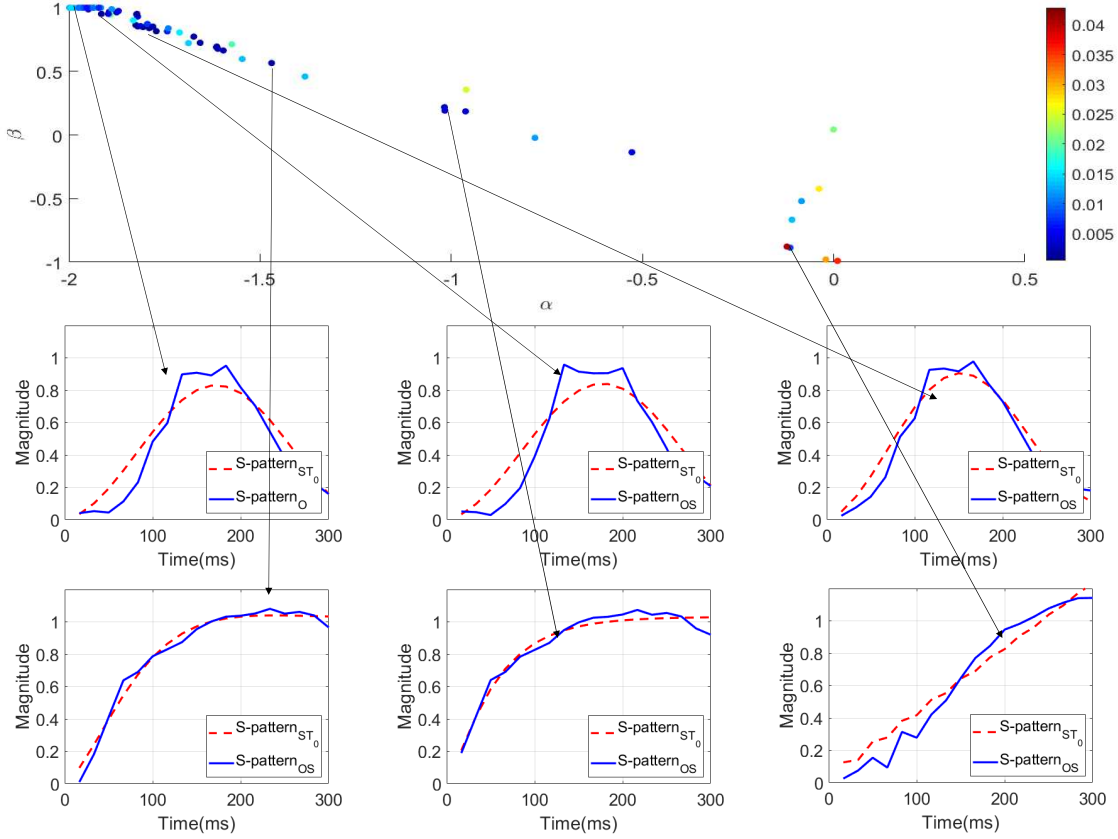


Figure 4-6: The distribution of  $(\alpha, \beta)$  is related to the curve shape of S-pattern. Each S-pattern ( $S\text{-pattern}_{OS}$ ) has been associated with its own identified Hammerstein model  $(\alpha, \beta, E_H)$ . The upper-left figure shows the distribution of  $(\alpha, \beta)$  ( $x$ :  $\alpha$ ,  $y$ :  $\beta$ ) and the associated error:  $E_H$  (heat map).  $(\alpha, \beta)$  densely distributes at the top-left corner with a small error. In the six below images, the blue curve means the original S-patterns ( $S\text{-pattern}_{OS}$ ). Then, based on the estimated Hammerstein model with original  $(\alpha, \beta)$ , the synthesized S-patterns ( $S\text{-pattern}_{ST_0}$ ) are generated (red curve). The curve shape of S-patterns in these six figures vary along with the change of  $(\alpha, \beta)$ . The first three curve images correspond to the densely distributed region of  $(\alpha, \beta)$ . The corresponding  $(\alpha, \beta)$  for the last three curve images are far from the upper-left region. They have different curve shapes compared with the first three. The distribution of  $(\alpha, \beta)$  is associated with the dynamic property of ME (shape of S-pattern). Hence, we would be able to both filter wrongly-labeled  $S\text{-pattern}_O$  using  $E_H$  values and also synthesize virtual S-patterns based on the value range of  $(\alpha, \beta)$ .

#### 4.2.4 Configurations for LTP Filtering and S-pattern Synthesizing

LTP filtering and S-pattern synthesizing can be performed thanks to the analysis in subsection 4.2.3. Firstly, S-patterns, which are wrongly labeled, can be filtered by  $E_H$ . Secondly, the  $(\alpha, \beta)$  distribution allows to define the  $(\alpha, \beta)$  value range for synthesizing reliable S-patterns by Hammerstein model.

For each database, the S-patterns after the whole LTP selection process of subsection 3.3.1 (S-pattern<sub>OS</sub>) are used to get the distribution of  $(\alpha, \beta)$  and their corresponding estimation error  $E_H$ . S-patterns<sub>OS</sub> are the most optimal LTPs for micro-expression thanks to the strict selection process. Hence, the  $(\alpha, \beta)$  distribution of S-patterns<sub>OS</sub> can be used as a reference for following LTP filtering and S-pattern Synthesizing. More precisely, we calculate the mean values and standard deviations:  $\bar{\alpha}$ ,  $\bar{\beta}$ ,  $\bar{E}_H$ ,  $\sigma_\alpha$  and  $\sigma_\beta$  for LTP filtering and S-pattern Synthesizing. Figure 4-7 shows the deduction process.

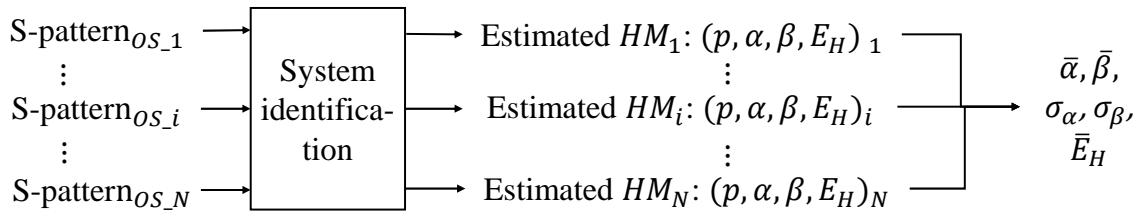


Figure 4-7: Parameter configuration for LTP filtering and S-pattern synthesizing. For one database, each selected S-pattern is treated separately to estimate its specific Hammerstein model. Based on these obtained data, the mean value  $(\bar{\alpha}, \bar{\beta}, \bar{E}_H)$  and the standard deviation  $(\sigma_\alpha, \sigma_\beta)$  can be calculated.

In this section, we find that the parameters  $(\alpha, \beta)$  of the linear module and the estimation error  $E_H$  are associate with the dynamic property of micro-expressions (curve shape of S-pattern). The LTP filtering can use  $\bar{E}_H$  to filter un-reliable S-patterns (section 4.3). Meanwhile, the  $\bar{\alpha}$ ,  $\bar{\beta}$ ,  $\sigma_\alpha$  and  $\sigma_\beta$  can be utilized for S-pattern synthesizing (section 4.4).

### 4.3 LTP Filtering

Based on the previous analysis, a filtering processing using Hammerstein model (LTP filtering) is proposed in this subsection. This process replaces the LTP pattern selection in the training step of local classification (see Figure 4-1 at the beginning of this chapter). LTP filtering allows to conserve more reliable S-patterns for training.

LTP filtering process use the estimation error  $E_H$  to filter S-patterns which are wrongly labeled. The detailed process is presented as follows. For each LTP which is labeled as S-pattern, its mean squared error of the estimation ( $E_H$ ) for Hammerstein model can be obtained by system identification. Then,  $E_H$  is compared with a threshold of error value  $T_E$  to filter LTP patterns. The value assignment of  $T_E$  depends on  $\bar{E}_H$  that is defined in subsection 4.2.4. The larger  $T_E$  is, the more LTPs are conserved and treated as qualified S-pattern. A simple flow chart for LTP filtering process is shown in Figure 4-8

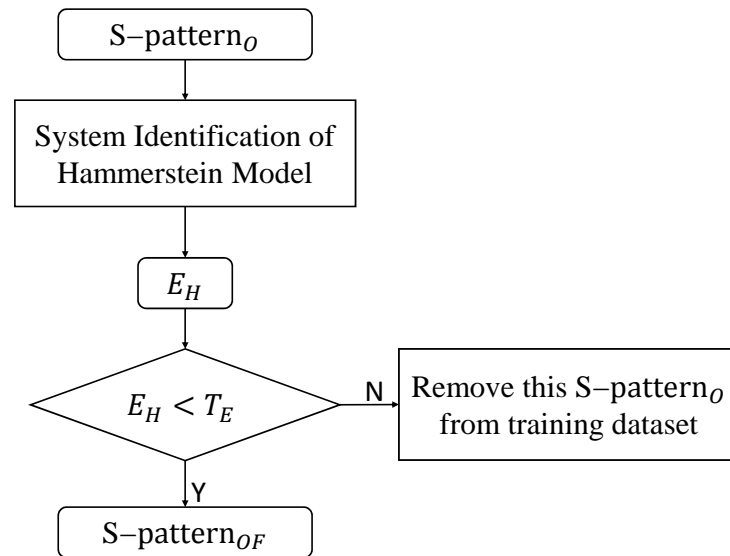


Figure 4-8: Flow chart of LTP filtering process. The original S-pattern<sub>O</sub> dataset may contain some wrongly-labeled samples. Each original S-pattern (S-pattern<sub>O</sub>) passes through the system identification to obtain its estimation error  $E_H$ . By comparing with the threshold  $T_E$ , the S-pattern<sub>O</sub> is decided to be kept as S-pattern after LTP filtering (S-pattern<sub>OF</sub>) or be removed from training dataset.

Figure 4-9 shows 4 examples of LTPs (labeled as S-pattern) who are not in accordance with the ideal S-pattern. They are removed from the training set by LTP filtering. The

movements which do not begin at real onset frame or are not similar to ME variation are eliminated efficiently. In this way, reliable S-patterns for training are conserved with a large amount. For example, for CASME I database, the amount of S-patterns after filtering is 2781, which is much larger than that of basic LTP pattern selection of subsection 3.3.1 (78). Indeed, in LTP pattern selection process, the curve shape is filtered by expert rules: the curve is raised up in first 150ms, and then the curve stabilizes or goes down. Yet, concerning LTP filtering, we 'learn' the curve shape with Hammerstein model. In other words, we create a selection by  $E_H$ : if the  $E_H$  of an S-pattern $_O$  is smaller than the threshold  $T_E$ , this S-pattern $_O$  is conserve as S-pattern $_{OF}$ ; otherwise, S-pattern $_O$  is deleted from the training dataset. **Hence, the data-scale can be increased using LTP filtering with Hammerstein model while the ability of distinguishing ME is maintained.**

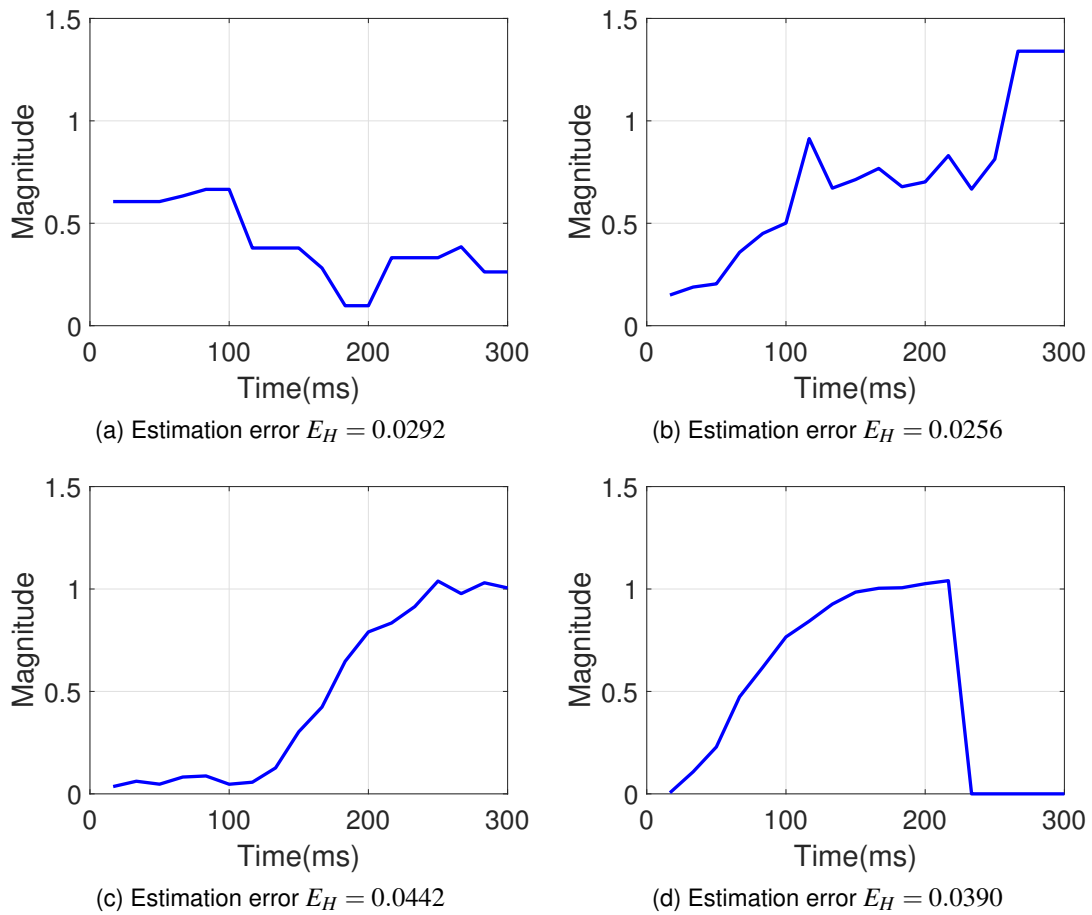


Figure 4-9: Examples of eliminated real LTPs (labeled as S-patterns) after S-pattern filtering. The threshold  $T_E$  is set to 0.0250. The curve shape in 4-9a represents a movement which begins to fade out. 4-9b and 4-9c show facial movements which are about to begin. 4-9d is the movement at the end of video sequence. These patterns are removed from training set by LTP filtering.

## 4.4 S-pattern Synthesizing

This section aims at increasing even more the number of S-patterns for the training of the classifier.

Reliable S-patterns are synthesized by Hammerstein model. We call them S-pattern<sub>ST</sub>. As introduced in sub-section 4.1 and 4.2, S-patterns with different curve shapes can be generated by Hammerstein model based on the variation of  $(\alpha, \beta)$ . Thus, depending on the parameters:  $\bar{\alpha}$ ,  $\bar{\beta}$ ,  $\sigma_\alpha$  and  $\sigma_\beta$  of the distribution for S-patterns,  $n$  sets of  $(\alpha_i, \beta_i)$  for each original S-pattern after LTP filtering (S-pattern<sub>OF</sub>) are generated from a normal distribution as follows:

$$\alpha_i = \bar{\alpha} + \sigma_\alpha \times R_i, \quad (4.2)$$

$$\beta_i = \bar{\beta} + \sigma_\beta \times R_i \quad (4.3)$$

where  $R_i$  is a value drawn from the standard normal distribution,  $i = 1, \dots, n$ . The  $(\alpha_i, \beta_i)$  are close to the most densely distributed region of initial distribution. Thus, it can be used to construct the Hammerstein model and synthesize representative S-pattern<sub>ST<sub>*i*</sub></sub>. For the sake of reproducibility of the algorithm, a flow chart is illustrated in Figure 4-10.

Figure 4-11 shows an example for S-pattern synthesizing. **Once the generation multiple  $n$  is defined,  $n$  S-patterns<sub>ST</sub> can be synthesized based on  $n$  sets of  $(\alpha, \beta)$ .**

After the S-pattern synthesizing, the S-patterns<sub>OF</sub> and their corresponding S-patterns<sub>ST</sub> are concatenated for a bigger training dataset and are used to train the SVM classifier. After the spatial and temporal fusion (subsection 3.3.3), we get the final spotting result. In section 5.4 of Chapter 5, we will prove that the data augmentation by S-pattern synthesizing improve the spotting performance.



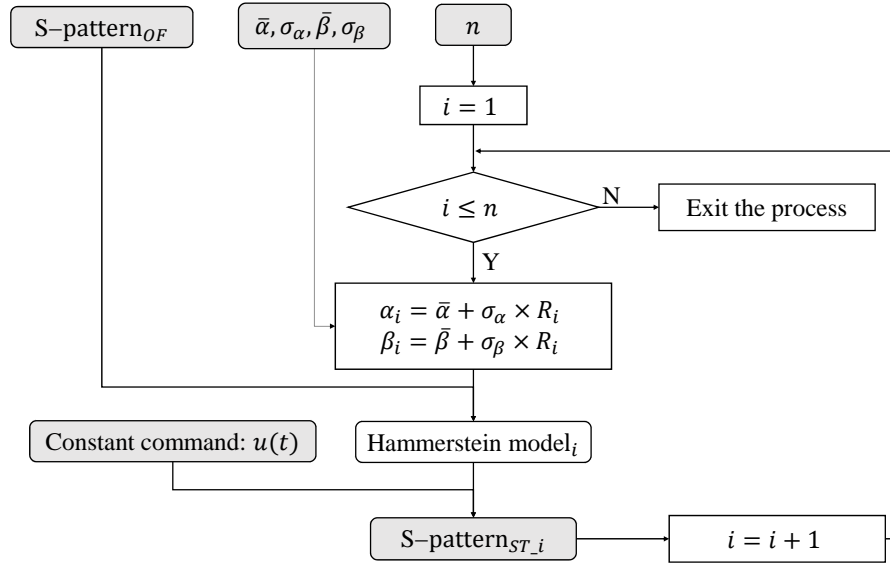


Figure 4-10: Flow chart of S-pattern synthesizing. The number of generation loops is defined by  $n$ . Once the  $(\alpha_i, \beta_i)$  is determined, along with the  $S\text{-pattern}_{OF}$ , the specific Hammerstein model is constructed for synthesizing.  $S\text{-pattern}_{OF}$  is needed in this step because it helps to identify the parameters  $p$  in the non-linear module. Then the  $S\text{-pattern}_{ST_i}$  is synthesized by the Hammerstein model $_i$  whose input is the constant command  $u(t)$ .

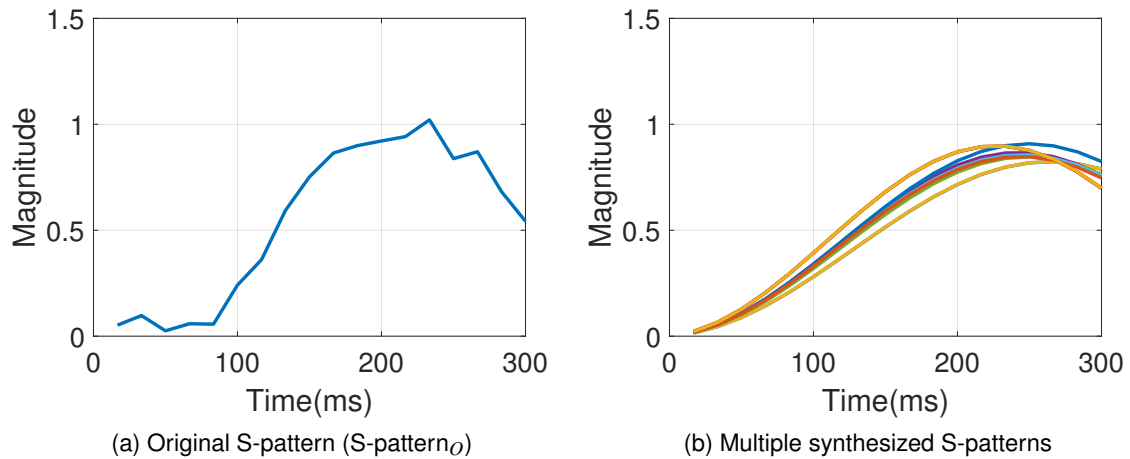


Figure 4-11: Example of 1 original S-pattern ( $S\text{-pattern}_O$ ) and 10 S-patterns generated by Hammerstein model ( $S\text{-patterns}_{ST}$ ). Depending on the  $S\text{-pattern}_O$ , we can generate  $n$  times similar  $S\text{-patterns}_{ST}$  for data augmentation.

## 4.5 Conclusion

In this chapter, we propose to use Hammerstein model for the data augmentation of the training set. With LTP filtering and S-pattern synthesizing, our method conserves a large amount of reliable S-patterns, increases the size of S-pattern dataset for training stage of machine learning, and therefore improves the spotting performance.

The following chapter will presents the experimental results. The comparison with the state of arts method proves the efficiency of our method.



# Chapter 5

## Experimental Results

This chapter aims at proving the efficiency of our whole method. The method is compared with the state of art (SOA) method: LBP- $\chi^2$ -distance method [91] (subsection 5.2.1 and 5.4.1).

The chapter is organized based on our four contributions, as illustrated in Figure 5-1.

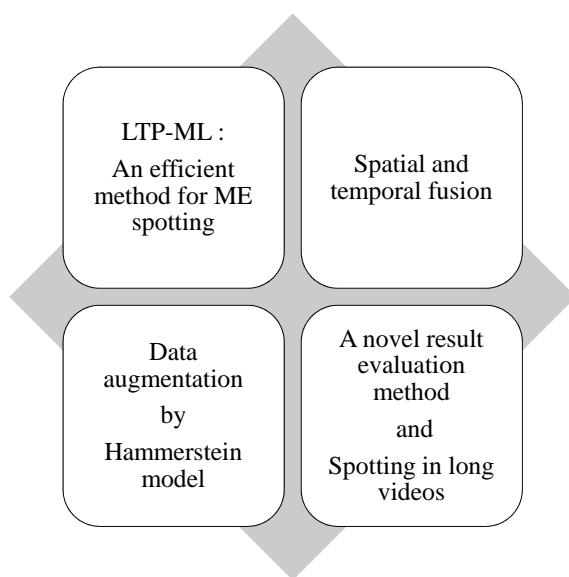


Figure 5-1: Organization of Chapter5 based on our four contributions.

One of our main contribution is to **spot ME by classifying a novel relevant feature: local temporal pattern**. In order to demonstrate the relevancy of the LTP pattern, subsection 5.2.2 compares it with another common used temporal feature (LBP-TOP). Subsection 5.2.3 proves the generality of our proposed LTP feature among different databases.

As we mention that S-pattern is identical for all kinds of emotions, the spotting performance per emotion and the statistic analysis of LTP for different emotions are investigated to prove the theory (subsection 5.2.4). Then, we analyse parameters in two sub-processes: extraction of ROI (Subsection 5.2.5) and PCA in LTP computation (Subsection 5.2.6). The analysis allows to find the optimal ROI setting and to prove the effectiveness of PCA.

Concerning our second contribution, **the spatial and temporal fusion** is investigated to show its capacity to differentiate ME from other movements (Subsection 5.3.1). As well, the impact of threshold values in the fusion process are analyzed to find the optimal parameters (Subsection 5.3.2).

Another main contribution of our process is **data augmentation using HM**. The comparison between our method with and without Hammerstein model shows that the data augmentation improves the spotting performance. To show the effectiveness of S-pattern synthesizing by Hammerstein model, the method is compared with GAN (Subsection 5.4.2). The analysis in subsection 5.4.3 shows the impact of LTP filtering and S-pattern synthesizing on the entire process. Meanwhile, in order to find the optimal threshold value for estimation error ( $T_E$ ) and generation multiple  $n$  for ME spotting, the impact of these parameters on two sub-processes are investigated respectively (subsection 5.4.4 and 5.4.5). Finally, the analyze of different distribution models of  $(\alpha, \beta)$  shows that the generation amount of S-pattern matters more than the choice of distribution model.

Finally, our contribution of **the novel result evaluation method and metrics** are introduced in section 5.1.4, and section 5.5 shows the results of **spotting micro-expression in long videos** utilizing our method.

The databases, the method for comparison, configurations and metrics for the experiments are firstly introduced in subsection 5.1. In this chapter, our method without and with Hammerstein model is mentioned as LTP-ML and LTP-SpFS respectively.

## 5.1 Databases, Comparison Method, Experimental Configuration and Metrics

In this section, as we need to perform our method on video samples to verify its feasibility, four databases are firstly introduced (subsection 5.1.1). Then in order to compare our method with the state-of-arts (SOA) method, LBP- $\chi^2$ -distance that is most commonly used for spotting method comparison is presented (subsection 5.1.2). For the reproducibility of our method, the configurations of experiments are listed in subsection 5.1.3. Finally, in order to evaluate the performance of our method, the result evaluation method and metrics for micro-expression spotting are introduced in subsection 5.1.4.

### 5.1.1 Databases

Four public spontaneous micro-expression databases are utilized for our experiments. CASME I and CASME II are two databases with short video sequences. They are chosen because most articles used these two databases to evaluate the result. Meanwhile, CAS(ME)<sup>2</sup> and SAMM the only two databases which contain long video samples. We use these two databases to test our method in long video situation. The detailed information is introduced in following paragraphs.

#### Databases with Short Videos

Experiments are firstly performed on two spontaneous ME databases: CASME I [137] and CASME II [137]. The MEs in these two databases are labeled with reliable ground truth, including the temporal location of the onset, apex and offset of the ME. In addition, the AU information is given for each video, permitting to create our ground truth for recognizing the LTP.

**CASME I:** The database contains 19 subjects, 177 videos and 197 MEs (some videos have several MEs). All videos are at a frequency of 60 fps. There are two sections : section A and section B because of two lighting conditions and also two different resolutions. In section A, there are 7 subjects, 96 ME sequence, and the resolution is  $720 \times 1280$ . In

section B, there are 13 subjects, 101 sequences of MEs and the resolution is  $640 \times 480$ .

**CASME II:** There are 26 subjects and 255 ME sequences. All CASME II videos are at 200 fps to retain more facial information and the resolution is  $640 \times 480$ .

All ME sequences in CASME I and CASME II are used in the experiment. The main parameters of these two databases are listed in Table 5.1.

### Databases with Long Videos

**CAS(ME)<sup>2</sup> [104]:** In the part A of CAS(ME)<sup>2</sup> database, there are 22 subjects and 87 long videos. The average duration is 148s. The facial movements are classified as macro- and micro-expressions. The video samples may contain multiple macro or micro facial expressions. The onset, apex, offset index for these expressions are given in an excel file. In addition, the eye blinks are labeled with onset and offset time.

**SAMM database [19]:** In SAMM database, there are 32 subjects and each has 7 videos. The average length of the videos is 35.3s. For this challenge, we focus on 79 videos, each contains one/multiple micro-movements, with a total of 159 micro-movements. The index of onset, apex and offset frames of micro-movements are outlined in the ground truth excel file. The micro-movements interval is from onset frame to offset frame. In this database, all the micro-movements are labeled. Thus, the spotted frames can indicate not only micro-expression but also other facial movements, such as eye blinks.

### 5.1.2 SOA Method For Comparison: LBP- $\chi^2$ -distance

The LBP-Chi-square-distance method (LBP- $\chi^2$ ) was first proposed by Moilanen et al. in 2014 [91]. The pipeline of this method is shown in Figure 5-2. We use this method to compare with, because it is the one that is most commonly used in other articles to compare with their own ME spotting results. However, most methods [83, 65] evaluate their results using ROC and AUC metrics. In our method, these metrics are not suitable since there is no valuable parameter to adjust. Another point is that, the results presented in [91] consider that the eye blinking is a true positive, which is not the case of the ground truth in the databases. As a result, we re-implement the method from the article and succeed to

achieve the same level of spotting rate. As the configurations of the experiment were not clearly indicated in the article, the setting of this method in our article will be presented in following sub-section.

### 5.1.3 Parameters Configuration

**LBP- $\chi^2$** : The configuration of LBP- $\chi^2$  method is based on the descriptions of the three articles: [91], [65] and [111]. The face is divided into 36 blocks with an overlap. The overlap rates in the direction of X and Y are 0.2 and 0.3 respectively. A uniform mapping is applied for the LBP feature extraction from the block, the radius  $r$  is set to  $r = 3$ , and the number of neighboring points  $p$  is set to  $p = 8$ . The  $\chi^2$  distances of the current frame are computed in an  $K$  interval. The  $K$  value for two databases are listed in Table 5.1. The ground truth of LBP- $\chi^2$  is put in the range of  $[\text{onset} - K/2, \text{offset} + K/2]$ .

**LTP-ML and LTP-SpFS**: For our method, 12 ROIs are selected at the eyebrows, the nose and the mouth contour, as shown in Figure 3-2. The size of the time interval  $K$  corresponds to 300ms according to the fps of each database as shown in Table 5.1. 300ms is the average duration of a ME. Training and recognition are performed using the software Lib-SVM with linear kernel [16]. Since the dataset is very unbalanced, non-ME frames are sampled by 1 out of 8 for SVM training stage. The parameter of cost  $c$  for SVM training is set to 5 and the weight  $w$  for each class are set to 1 and 2.5 respectively. All frames are considered in the testing stage. The results are obtained by LOSubOCV.

Since our method detects the special pattern of the onset of local facial movement, the optimal condition is to detect patterns in the interval of  $[\text{onset} - K/3, \text{onset}]$  and in the meantime in  $[\text{apex} - K/3, \text{apex}]$ . Thus, the ground truth of our method is defined by adding a  $K/3$  shift to that of LBP- $\chi^2$ , i.e.  $[\text{onset} - K/3 - K/2, \text{offset} - K/3 + K/2]$ .

### 5.1.4 Result Evaluation Method and Metrics

As introduced in chapter 2, there is no consistency of result evaluation methods for micro-expression spotting. To evaluate our method and to compare it with other methods, we evaluate the spotting result per frame and per interval, as introduced in following para-



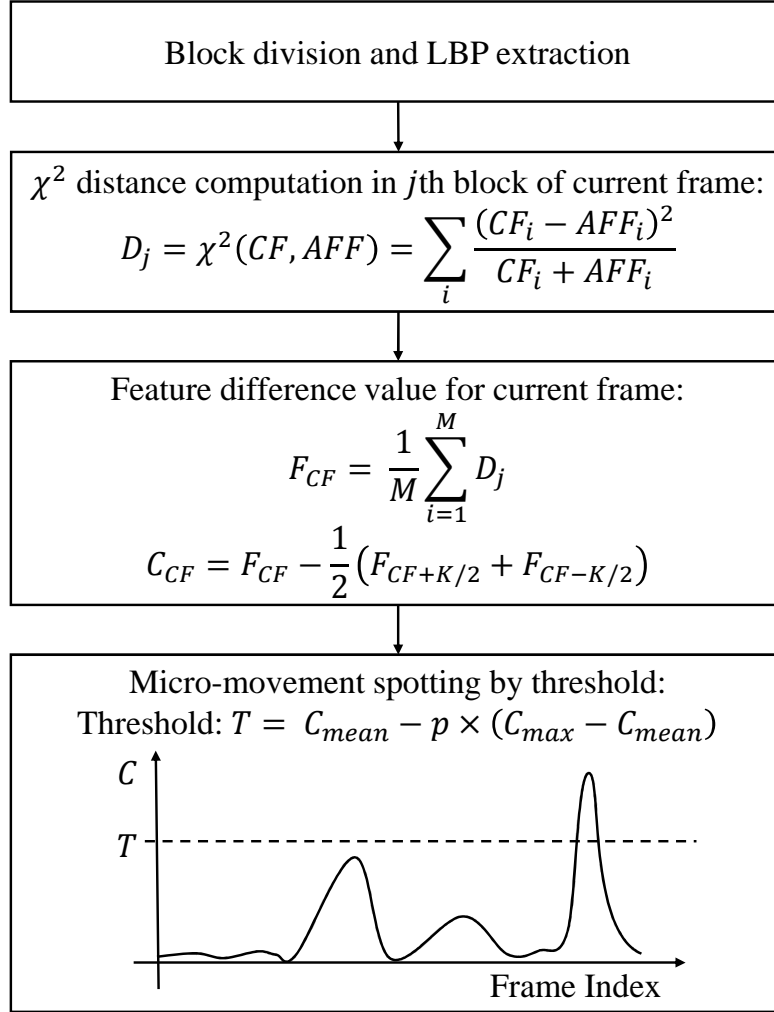


Figure 5-2: Baseline method -LBP  $\chi^2$ -distance difference. In the first step, the face image is divided into several blocks. Then LBP is computed per pixel. The feature for the  $j$ th block on current frame (CF) is the LBP histogram per ROI after a normalisation. The second step is  $\chi^2$ -distance computation.  $i$  is the  $i$ th bin in the histogram. AFF (Average feature frame) means the feature vector representing the average of tail frame and head frame, where tail frame is  $K/2$ th frame before the CF, head frame is  $K/2$ th frame after the CF. The third step is to obtain the final feature difference value  $C_{CF}$  for current frame.  $F$  is obtained by the first  $M$  blocks with the biggest feature difference values. The fourth step spots micro-expression by setting a threshold, where  $p$  is an empirical data.

Table 5.1: Main parameters and experiment parameters configuration for CASME I (Section A and B), CASME II, CAS(ME)<sup>2</sup> and SAMM.

Database	Subject	ME sequence	FPS	K
CASME I-A	7	96	60	18
CASME I-B	12	101	60	18
CASME II	26	255	200	60
CAS(ME) <sup>2</sup>	22	97	30	9
SAMM	32	79	200	60

graphs. The evaluation method per interval is proposed by MEGC2019 (Micro-expression Grand Challenge) [64], and we participated to the elaboration.

### Evaluating ME Spotting Result per Frame

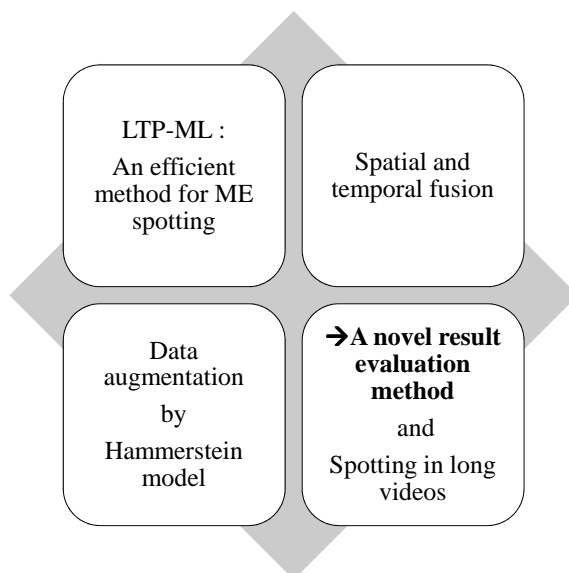
Concerning the result evaluation, we choose a measurement per frame. Indeed, LTPs are extracted from each frame. In addition, local classification and the first two steps of spatial and temporal fusion are performed per frame. Since F1-score is the most commonly used metric for the evaluation of MESR method using machine learning, it is chosen to confirm the effectiveness of our method. In the following experiments, if we do not mention specifically, the F1-score is the F1-score per frame by default. **F1-score** can be calculated based on *recall* and *precision*. Here are the definitions of these two metric for our results:

$$recall = TPR = \frac{\text{All spotted ME frames}}{\text{All ME frames in database}} \quad (5.1)$$

$$precision = \frac{\text{All spotted ME frames}}{\text{All spotted frames in database}} \quad (5.2)$$

**Yet, one of the challenges in ME spotting is to have a low FPR (False Positive Rate). That is why we consider that having a FPR over 0.2 is not acceptable.**

## Evaluating ME Spotting Result per Interval



We participated to the ME spotting task of the MEGC2019 (Micro-expression Grand Challenge) [64] and proposed **a new result evaluation method and metrics** through an international collaboration. It is an **F1-score by interval**.

The annotation of micro-expression is not always accurate. As it is the ground truth for ME spotting evaluation, the uncertainty influences the evaluation result. To avoid this inaccuracy, we propose to evaluate the spotting result per interval. Moreover, since the distribution of ME and non-ME sequences is not balanced, F1-score is utilized.

There are three evaluation steps used to compare the performance of the spotting tasks:

**1. True positive in one video definition** Supposing there are  $m$  micro-expressions in the video, and  $n$  intervals are spotted. The result of this spotted interval  $W_{spotted}$  is considered as *true positive (TP)* if it fits the following condition:

$$\frac{W_{spotted} \cap W_{groundTruth}}{W_{spotted} \cup W_{groundTruth}} \geq k \quad (5.3)$$

where  $k$  is set to 0.5,  $W_{groundTruth}$  represents the micro-expression interval (onset-offset). Otherwise, the spotted interval is regarded as *false positive (FP)*.

**2. Result evaluation in one video** Supposing the number of *TP* in one video is  $a$  ( $a \leq m$  and  $a \leq n$ ), then  $FP = n - a$ , *false positive (FN)* =  $m - a$ , the *Recall*, *Precision* and *F1-score*

are defined:

$$Recall = \frac{a}{m}, Precision = \frac{a}{n} \quad (5.4)$$

$$F - score = \frac{2TP}{2TP + FP + FN} = \frac{2a}{m + n} \quad (5.5)$$

In practical, these metrics might not be suitable for some videos, as there exist the following situations on a single video:

- The test video does not have micro-expression sequences, thus,  $m = 0$ , the denominator of recall will be zeros.
- The spotting method does not spot any intervals. The denominator of precision will be zeros since  $n = 0$ .
- If there are two spotting methods, Method<sub>1</sub> spots  $p$  intervals and Method<sub>2</sub> spots  $q$  intervals, and  $p \leq q$ . Supposing for both methods that the number of true positive is 0, thus the metrics (*recall*, *precision* or *F1-score*) values both equal to zeros. However, in fact, the Method<sub>1</sub> spots less false positives than Method<sub>2</sub>.

Considering these situations, for a single video, we propose to record the result in terms of  $TP$ ,  $FP$  and  $FN$ . For performance comparison, we produce a final calculation of other metrics for the entire database.

**3. Evaluation for entire database** Supposing that in the entire database, there are  $V$  videos and  $M$  micro-expression sequences, and the method spot  $N$  intervals in total. The database could be considered as one long video. Thus, the metrics for entire database can be calculated by:

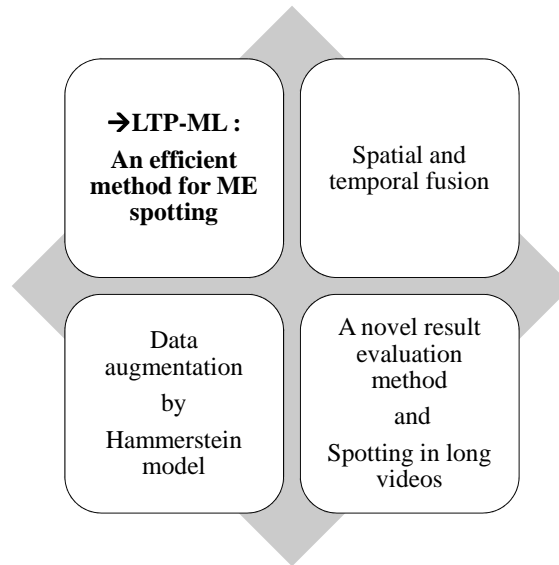
$$Recall_D = \frac{\sum_{i=1}^V a_i}{\sum_{i=1}^V m_i} = \frac{A}{M} \quad (5.6)$$

$$Precision_D = \frac{\sum_{i=1}^V a_i}{\sum_{i=1}^V n_i} = \frac{A}{N} \quad (5.7)$$

$$F1 - score_D = \frac{2 \times (Recall_D \times Precision_D)}{Recall_D + Precision_D} \quad (5.8)$$

The final results by different methods would be evaluated by *F1-score* since it considers the both *recall* and *precision*.

## 5.2 LTP-ML : An Efficient Method for ME Spotting



This section shows the first contribution of my thesis. The performance of LTP-ML is evaluated by the comparison with the LBP- $\chi^2$ -distance method using two public databases CASME I and CASME II. The results are analyzed per frames. Then, we demonstrate the relevancy of our proposed LTP feature for micro-expression spotting. We also prove the generalization of LTP for different databases. Parameters for ROI extraction are also discussed in this section. Meantime, PCA is proven to be suitable for our method by the comparison with Auto-Encoder and GPLVM. In addition, we evaluate the spotting performance per emotion and perform the statistical analysis on S-patterns for different emotions, which shows that our LTP pattern is identical for all emotions.

### 5.2.1 LTP-ML Outperforms SOA LBP- $\chi^2$ Method

Compared with the LBP- $\chi^2$  method, LTP emphasizes on the local temporal facial deformation. Classifying LTPs by machine learning also enhances the ability of differentiating ME from other facial movements. In this subsection, the spotting results of LTP-ML are compared with LBP- $\chi^2$ .

Since the original LBP- $\chi^2$  process does not perform any merge after the peak detection, the results obtained by our method are firstly compared with LBP- $\chi^2$  without the merge

process. The results are listed in Table 5.2. All the results have FPR lower than 0.2, which means that the spotting method is acceptable. LTP-ML without the merge process outperforms LBP- $\chi^2$  in term of F1-score for each database, because the TPR of our method is higher than 20%. Even though FPR is slightly higher, the value of F1-score is raised due to the increase of TPR (True Positive Rate).

Table 5.2: Result evaluation and comparison for LTP-ML and LBP- $\chi^2$ . LTP-ML (without merge) can spot more TP frames with an acceptable FPR. In addition, LTP-ML outperforms state-of-art LBP- $\chi^2$  method in terms of F1-score in both cases: with or without merge process. Merge process helps to reduce the true negatives. (F1-score<sub>fr</sub>: F1-score per frame)

Database	Method	Without merge			With merge		
		TPR	FPR	F1-score <sub>fr</sub>	TPR	FPR	F1-score <sub>fr</sub>
CASME I-A	LBP- $\chi^2$	0.05	0.02	0.09	0.13	0.05	0.20
	LTP-ML	0.23	0.08	<b>0.30</b>	0.37	0.12	<b>0.40</b>
CASME I-B	LBP- $\chi^2$	0.07	0.02	0.12	0.18	0.05	0.24
	LTP-ML	0.22	0.05	<b>0.30</b>	0.34	0.09	<b>0.38</b>
CASME II	LBP- $\chi^2$	0.09	0.02	0.16	0.24	0.08	0.35
	LTP-ML	0.24	0.09	<b>0.35</b>	0.55	0.19	<b>0.59</b>

Due to the absence of merging the detected frames into a temporal interval, the TPR is not very high. In our proposed method, to reduce false negatives, the spotted frames pass through the merge process. For a fair comparison, we perform the same merge process on the LBP- $\chi^2$  method. Table 5.2 also shows the final spotting result with merge. As expected, the merge process improves the spotting performance for both methods. Our LTP-ML method still performs better than LBP- $\chi^2$ . A focus on the merge process is detailed in subsection 5.3.

## 5.2.2 Relevancy of LTP Compared with LBP-TOP for ME Spotting

To prove the relevancy of our proposed LTP pattern, another commonly used feature (LBP-TOP) is computed in CASME I-A. LBP-TOP is extracted by a sliding window per frame, and the configuration of block division is inspired by [111]. Figure 5-3 shows an example of LBP-TOP extraction process per ROI. Then, the same machine learning and fusion process

as LTP-ML are performed on LBP-TOP. Experiments are performed on an Intel (R) Core (TM) i7 (-5820K CPU @ 3.3GHz) PC with 64GB RAM using Matlab R2018b. The average computation time for one video with 200 frames is counted for comparison. Table 5.3 lists the spotting results and the computation time for LBP-TOP and LTP. Our LTP allows to spot ME more accurately by extracting the main temporal variation from local regions. Moreover, our method takes less computation time than LBP-TOP, which is known to be efficient but time-consuming. LTP is faster thanks to a smaller feature dimension.

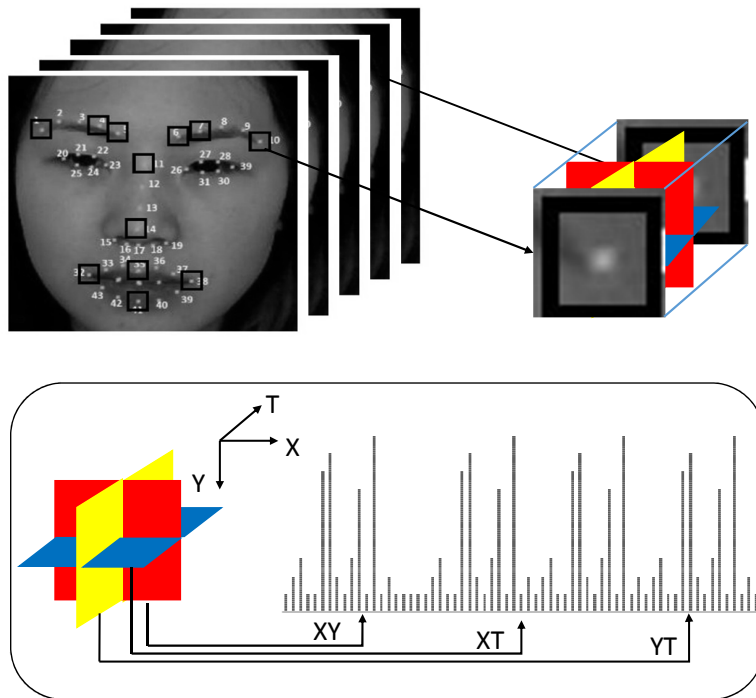


Figure 5-3: LBP-TOP extraction per ROI. LBP feature is extracted from three orthogonal planes: xy, xt and yt.

Table 5.3: LTP outperforms LBP-TOP on both spotting accuracy and computation time.  $Time_{extract}$  means the average time for feature extraction per one video sample with 200 frames, and  $Time_{svm}$  means the average time for SVM training and classifying per video. (CASME I-A)

	F1-score <sub>fr</sub>	$Time_{extract}$ (s)	$Time_{svm}$ (s)
LBP-TOP	0.30	15.00	9.89
LTP	<b>0.40</b>	<b>0.79</b>	<b>0.05</b>

### 5.2.3 Generalization of LTP for Different Databases

To investigate the generalization of our proposed feature: LTP among different database, we performed the cross database validation. The SVM model is firstly trained in one database, then predict the labels in another database. As listed in Table 5.4, the result proves the generalization of our proposed LTP pattern. The SVM model trained on CASME II can obtain a better classification result on CASME I-A and CASME I-B than their self-validation in the same database. It is because that CASME II has more S-patterns to train the classifier. Then, as the S-patterns that are extracted from different databases have the same property, the classifier trained by CASME II is able to spot micro-expressions in CASME I-A and CASME I-B.

Table 5.4: Cross database ME spotting performance shows the generality of LTP among different databases. The cross-database result is evaluated by F1-score<sub>fr</sub>. The experiments in the same database is performed by leave-one subject-out cross validation.

Train \ Test	CASME I-A	CASME I-B	CASME II
CASME I-A	0.40	0.40	0.40
CASME I-B	0.37	0.38	<b>0.42</b>
CASME II	0.52	0.41	<b>0.59</b>

### 5.2.4 Unique S-pattern for All Emotions

In our method, we treat all the regions of interest (ROIs) as undifferentiated. In other words, we make the assumption that the S-patterns are unique for all ROIs. To verify this assumption, we compare the F1-score per emotion in CASME I.

#### ME spotting on different emotions

The video samples in CASME I are labeled by emotion type and AU information. Table 5.5 lists the measurement per frame for CASME I. Despite of fear emotion with only one video sequence, there is nothing significant found by the analysis of F1-score measure. While the tense and surprise emotion are mostly linked to the AU1, AU2 and AU4 of eyebrow, the emotion of happiness and repression always lead to the mouth movement. The different



emotion links to different AU combination, which means the ROIs which contains ME movement are different for each emotion. Yet, the type of emotion does not influence the spotting result. These results confirm that our proposed S-pattern is similar regardless of the ROI and the ME type.

Table 5.5: Spotting results per emotion on CASME I

Emotion	CASME I-A		CASME I-B	
	NbVideo	F1-score <sub>fr</sub>	NbVideo	F1-score <sub>fr</sub>
Tense	48	0.27	23	0.33
Happiness	4	0.48	5	0.28
Repression	30	0.28	10	0.08
Disgust	4	0.30	42	0.41
Surprise	7	0.45	14	0.57
Contempt	4	0.18	6	0.58
Fear	1	0.55	1	0.34

### Statistic Analysis of S-pattern

This subsection proves the S-pattern is unique for micro-expressions and it has the similar pattern for different emotions. The maximal value of S-pattern ( $D_{max}$ ) and the slope of S-pattern curve ( $Slope_{150ms}$ ) are chosen for the statistic analysis. It is because the slope indicates the speed of the facial movement and the maximal values of S-pattern shows the movement intensity. The first part of this subsection shows the uniqueness of S-pattern for micro-expression, then the second part proves that S-patterns are identical for all emotions.

**S-pattern is Unique for Micro-expressions** As show in Table 5.6, S-pattern differs from other LTPs (non-S-patterns). The maximal value of S-patterns is around 1 while the non-S-patterns have a much smaller value. Because most frames are neutral face, they barely have movements, the LTPs for non-ME frames have a small value. In addition, since LTPs are normalized by the maximal original distance ( $\lambda_{max}$ ) in the first 150ms, the  $\overline{D_{max}}$  shows that most micro-expressions reach the maximal value in the first 150ms then the values begin to decline. If not, the average value will be bigger than the current one. Besides, the average curve slope for S-patterns is sharper than non-S-patterns, because micro-expression is a very brief facial movement.

Table 5.6: S-pattern differs from other LTP (non-S-patterns). The LTP is statistically analyzed by the average value and the standard deviation of the following two characteristics: the maximal value of S-pattern ( $D_{max}$ ) and the curve slope of S-pattern in first 150ms ( $Slope_{150ms}$ ).

Database	Feature	$\overline{D_{max}}$	$\sigma(D_{max})$	$\overline{Slope_{150ms}}$	$\sigma(Slope_{150ms})$
CASME I-A	S-pattern	<b>1.03</b>	0.13	<b>0.09</b>	0.03
	non-S-pattern	0.52	0.36	0.02	0.03
CASME I-B	S-pattern	<b>1.00</b>	0.10	<b>0.09</b>	0.03
	non-S-pattern	0.44	0.33	0.01	0.03

**S-pattern is Identical for Different Emotions** This subsection shows the statistical analysis of S-patterns for different emotions. The two sub-table in Table 5.7 listed the results for two kinds of emotion classifications. The maximal distance value and the curve shapes for different emotions are similar, and the deviation is small. In conclusion, S-pattern is identical for all emotions

Table 5.7: Unique S-pattern for different emotions. The S-pattern is statistically analyzed by the average value and the standard deviation of the following two characteristics: the maximal distance value in S-pattern ( $Dist_{max}$ ) and the curve slope of S-pattern in first 150ms ( $Slope_{150ms}$ ).

Database	Emotions	$\overline{Dist_{max}}$	$\sigma(Dist_{max})$	$\overline{Slope_{150ms}}$	$\sigma(Slope_{150ms})$
CASME I-A	Positive	1.15	0.09	0.11	0.01
	Negative	1.01	0.13	0.09	0.04
	Surprise	1.08	0.02	0.11	0.01
CASME I-B	Positive	1.11	0.11	0.08	0.02
	Negative	1.00	0.10	0.09	0.03
	Surprise	1.00	0.11	0.09	0.03

(a) S-patterns for 3 classes of emotions

Emotions	$\overline{Dist_{max}}$	$\sigma(Dist_{max})$	$\overline{Slope_{150ms}}$	$\sigma(Slope_{150ms})$
Happiness	1.11	0.11	0.08	0.02
Tense	0.99	0.10	0.09	0.03
Repression	1.04	0.07	0.07	0.05
Disgust	1.01	0.10	0.09	0.03
Contempt	0.98	0.09	0.08	0.03
Surprise	1.00	0.11	0.09	0.03

(b) S-patterns for 6 emotions

## 5.2.5 Impact of ROI Parameters on Spotting Results

There are two main parameters for ROI extraction process which influence the spotting performance: ROI size and ROI amount. In this section, the impact of these two parameters on our method is investigated to find the optimal setting for ROI extraction.

### Impact of ROI size on spotting results

The block length  $a$  of ROI is worth to study in order to find an appropriate size for feature extraction. The influence of the size on spotting result is analyzed in CASMEI [137]. The facial resolution of participants is around  $150 \times 190$ . The dataset of  $a$  for the experiment is [5, 10, 15, 20, 25, 30, 35, 40, 50, 70]. The spotting results depending on different ROI size are illustrated in Figure 5-4. As shown in this figure, spotting method with ROIs of  $20 \times 20$  (pixel<sup>2</sup>) has the best result. In this case,  $a = 1/5 \times L = 20$ , where  $L$  is the average inner eye corner distance for all video samples in CASME I. The ROI in this size contains sufficient local facial displacement, while avoiding the overlap of adjacent ROIs. Thus,  $a = 1/5 \times L$  is applied as an empirical setting for the pre-processing in all databases.

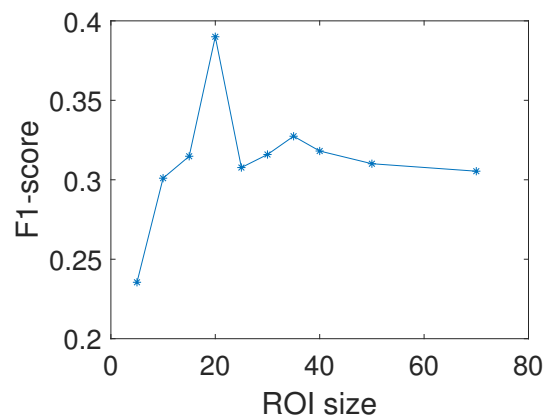


Figure 5-4: Influence of ROI size on Spotting performance. The F1-score increases along with the augmentation of ROI size before the region length reaches 20. It is because that the information in regions which are too small is not enough to represent the micro-expression local movement. Yet, the spotting performance is then affected when the ROI size gets larger than 20. More irrelevant information is included in the region for analysis. It raise the false positive in the final analysis.

## Impact of ROI amount on spotting results

The amount of ROIs is analyzed in this part, since it is related to the efficiency of our method. As shown in Table 5.8, two conditions of ROI amount are utilized. The first one is an entire selection, i.e. all 26 ROIs of eyebrows, nose and mouth are chosen for micro-expression spotting. And the second one is a partial selection (12 ROIs): ROI 1, 4, 5, 6, 7, 10 (eyebrows); ROI 32, 35, 38, 41 (mouth) and ROI 11, 14 (nose). These ROIs are the regions which contain the most evident muscle movement of micro-expression compared with other ROIs in the same area. Table 5.9 lists the spotting result of these two conditions. The F1-score for the entire selection is moderately improved since there are more samples for training. Yet, the false positive rate (FPR) is largely raised. Since the partial selection can get the proximate spotting result and a smaller FPR compared with the entire selection, the following processes are performed on only 12 ROIs for computation efficiency.

Table 5.8: Two situations of ROI index. The ROI index is annotated depending on detected facial landmarks.

Facial region	12 ROI index	26 ROI index
Eyebrows	1, 4, 5, 6, 7, 10	[1-10]
Nose	11, 14	11,12,13,14
Mouth	32, 35, 38, 41	[32-43]

Table 5.9: Spotting performance evaluation (F1-score) based on different amount of chosen ROIs.

Databases	CASMEI-A		CASMEI-B	
	FPR	F1-score <sub>fr</sub>	FPR	F1-score <sub>fr</sub>
12 ROIs	0.12	0.40	0.09	0.38
26 ROIs	0.20	0.41	0.17	0.40

## 5.2.6 PCA is More Suitable for Dimension Reduction in Our Method than Auto-Encoder and GPLVM

Auto-encoder is an emerging machine learning tool for dimension reduction. It is an unsupervised process with multiple-layer neural networks. Compared with PCA, the auto-encoder system has a stronger ability of extracting information, because the process also

conserves the non-linear characteristic of features. The auto-encoder system applied in our LTP extraction process is illustrated in figure 5-5.

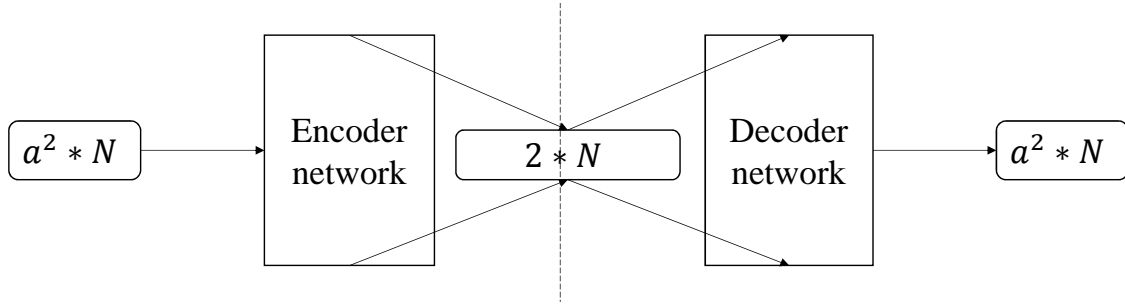


Figure 5-5: Auto-encoder system for dimension reduction of ROI sequence on time axis ( $N$ ). The auto-encoder system can encode the ROI sequence ( $a^2 * N$ ) into a 2D dimension point distribution, and then reconstruct the input by decoder process.  $a$  is the ROI width,  $a^2$  is the pixel amount in one ROI image, and  $N$  means the frame amount in this ROI sequence. The frame index represents the temporal relation in this video.

To obtain the 2D point set  $P_1, \dots, P_n$ , the ROI sequence in size of  $a^2 * N$  passes through an encoder. This model is the left part of the AE system shown in Figure 5-5. The system is trained by concatenating ROI sequences in one video, i.e. each video sample has its own specific trained encoder. The detailed schema is illustrated in Figure 5-6.

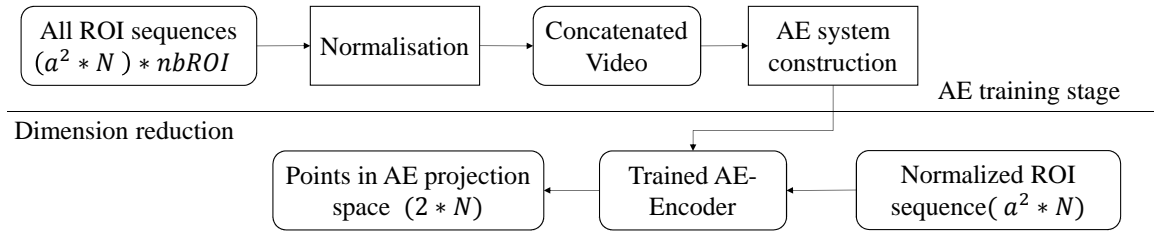


Figure 5-6: Dimension reduction schema by Auto-encoder. The process contains two stages: AE training stage and dimension reduction. For each video sample, all  $J$  ROI sequences are concatenated into one matrix in size of  $(a^2 * N) * J$ . To conserve the principal variation, the matrix is normalized for AE training. Then, the obtained encoder can extract the maximal movement on time axis of the chosen ROI sequence. Like PCA process, the dimension of the input data is reduced to 2 dimensions. Thus,  $N$  2D points which represent the ROI sequence can be obtained in AE projection space.

After analyzing the point distribution, we could find the similar variation pattern as LTP. An example of the comparison of LTPs obtained by PCA and AE is shown in Figure 5-7. Local temporal patterns have the same curve shape for the same ROI at the same moment.

Thus, the point set obtained by AE can be treated by the same process as PCA for further micro-expression spotting.

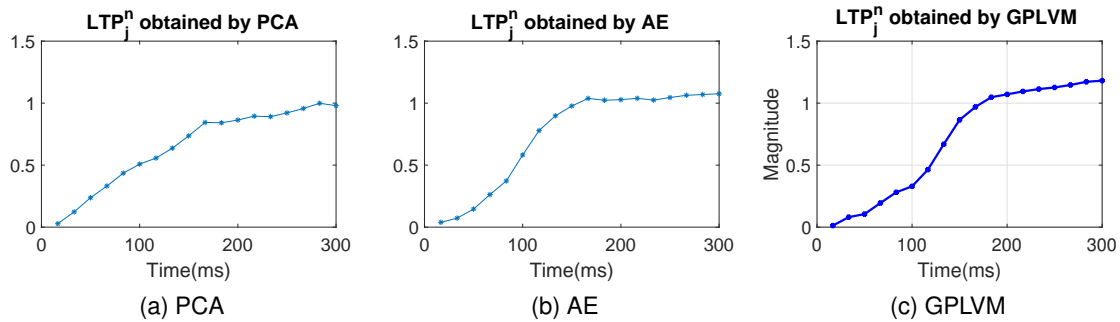


Figure 5-7: Comparison between LTP examples obtained by PCA, AE and GPLVM. For the same micro-expression sequence, these two dimension reduction methods can obtain similar LTP patterns. (CASMEI-A, sub1\_EP07\_10, ROI5)

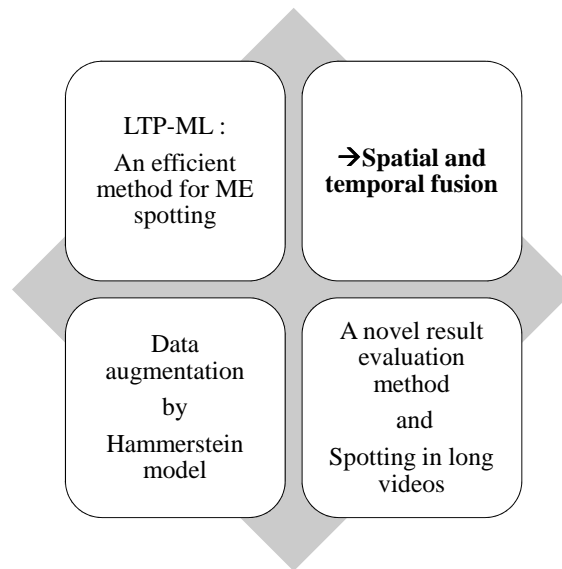
Table 5.10 shows the spotting result. The experiments are performed on an Intel(R) Core(TM) i7(-3740QM CPU @ 2.7GHz) PC with 8GB RAM using Pycharm. AE can conserve more useful movement in only 2 dimensions, and the spotting performance is slightly better than that of PCA. However, by comparing the time of feature extraction for the entire database of CASME I, we find that the AE process is time consuming. The computation time is almost 10 times of that of PCA. Hence, PCA is more suitable for feature extraction, since it is rapid and conserves sufficient movement information.

GPLVM (Gaussian Process Latent Variable Models) is also a frequently used method for dimension reduction. Figure 5-7 shows that the local temporal pattern of GPLVM also forms the S-pattern for micro-expression. We compare the spotting result using LTPs extracted by GPLVM and PCA, as shown in Table 5.10. The two process can get the similar spotting but GPLVM takes more computation time for feature extraction. Thus, for the purpose of efficiency, PCA is chosen in our method.

Table 5.10: Spotting performance evaluation after PCA and AE process. The F1-score of AE is slightly higher than that of PCA. And the GPLVM does not improve much the spotting performance. Yet, the time for feature extraction from the entire database by AE and GPLVM is much longer than PCA. PCA is more suitable for spotting micro-expressions in real time.

Databases	CASMEI-A		CASMEI-B	
	F1-score <sub>fr</sub>	Time <sub>extract</sub> (s)	F1-score <sub>fr</sub>	Time <sub>extract</sub> (s)
PCA	0.40	36.00	0.38	3.64
AE	0.42	230.4	0.41	36.36
GPLVM	0.33	172.8	0.40	121.2

## 5.3 Differentiating ME from Other Facial Movement by Spatial and Temporal Fusion



This section shows the second contribution of my thesis: the spatial and temporal fusion. As described in Chapter 3, the spatial and temporal fusion is one essential part of our method, and it is divided into three steps: local qualification, spatial fusion and merge process. We have already shown in Tabel 5.2 of Subsection 5.2.1 that the merge process improves the spotting result. In this section, we analyze the impact of each step on the spotting performance (subsection 5.3.1). Moreover, to optimize the fusion process, the impacts of two threshold values:  $T_{dist}$  and  $T_{CN}$  on the fusion process are investigated (subsection 5.3.2).

### 5.3.1 Impact of Each Step of the Fusion process on the Spotting Performance

Figure 5-8 presents a qualitative analysis. It illustrates an example of the contribution of each step. As shown in the second and third layer of the figure, the local qualification and the spatial fusion eliminate some detected peaks which do not fit the selection criteria. There is a risk that some TPs (True Positives) frames are deleted. Yet, in the fourth layer,



the merge process puts the neighboring detected frames into an interval who has appropriate length. The figure at the fifth layer illustrates the result with the three steps of spatial and temporal fusion. Frames that meet both the requirements of LQ and SF are conserved and merged into intervals.

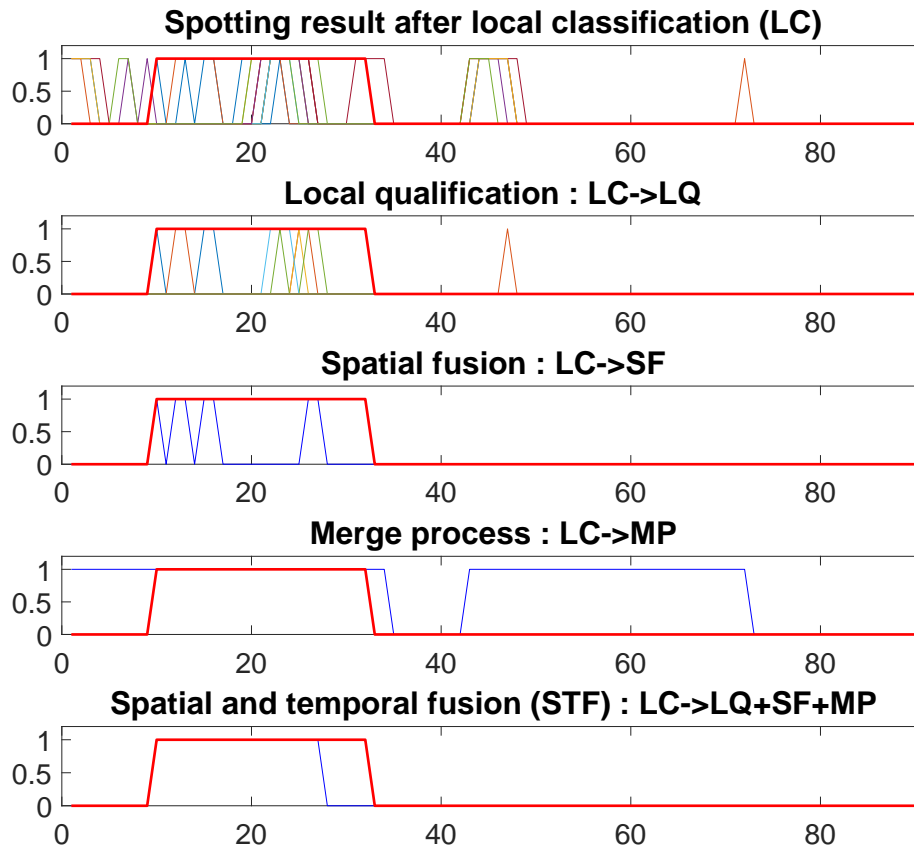


Figure 5-8: Example of global result obtained by each step in spatial and temporal fusion. The X axis is the frames index, the Y axis is the predicted label, and the red curve represents the ground truth for this video. As introduced in Figure 3-12, local classification and qualification are applied on ROIs. The first and second layers in the figure show the result obtained directly by the local classification (LC) and result after local qualification (LQ). The different colored lines represent the spotting results per ROI. The third and fourth layers give the global results after separately applying spatial fusion (SF) and merge process (MP) over the LC global result (blue curve). And the fifth layer is the final result after the three spatial and temporal fusion steps (STF). (CASME I\_Sub08\_EP12\_2\_1)

We also evaluate quantitatively the contribution of each step for CASME I-A (Table 5.11). In order to compare the local result of LC and LQ with the result of the other three steps, a naive global fusion is performed on LC and LQ: the frame is treated as ME if there is at least one ROI that spots S-patterns. The result in Table 5.11 shows that the

local qualification and spatial fusion can largely reduce the FP frames, while the number of TP also decreases. Conversely, the merge process increases the number of both TP and FP. Even though the F1-score of LC and MP are higher than other processes, their FPRs exceed 0.2. Therefore, the single process of LC or MP cannot be considered as an efficient ME spotting method. Yet, the combination of these three steps can reach to a balance: the F1-score almost remains the same while the FPR is acceptable and the TPR is slightly impacted.

Table 5.11: Analysis of each step for spatial and temporal fusion (STF) on CASME I-A. LC: local classification result; LQ: local qualification; SF: spatial fusion and MP: merge process. The decreasing of FP amount shows that LQ and SF process helps to reduce the irrelevant facial movements. More TP frames are spotted due to the merge process. A combination of these three steps keeps the spotting performance and largely reduces the false positives.

	TP	FP	F1-score <sub>fr</sub>	TPR	FPR
LC	1515	2943	0.40	0.50	0.25
LQ	712	<b>959</b>	0.30	0.23	<b>0.04</b>
SF	592	<b>701</b>	0.37	0.19	<b>0.05</b>
MP	<b>2123</b>	4903	0.42	<b>0.69</b>	0.41
STF	1138	1591	0.40	0.37	0.13

### 5.3.2 Impact of $T_{dist}$ and $T_{CN}$ on the Fusion Process

This subsection analyze the impact of two parameters of local qualifications:  $T_{dist}$  and  $T_{CN}$  on the fusion process.

Since the distance values in S-pattern are normalized, the threshold range of distance is set around 1 : [0.6, 0.7, 0.8, 0.9, 1]. Values larger than 1 are not used because they are too high to conserve useful S-patterns. As shown in Table 5.12, when the  $T_{dist}$  has a lower value, the algorithm spot more true positives (TPs). Yet, there are also too many false positives (FPs). When  $T_{dist}$  equals to 1, we get the smallest FP amount, and the spotting performance is not highly influenced. Also,  $T_{dist} = 1$  conserves the most ideal S-pattern according to our normalisation condition.

Table 5.12: Impact of  $T_{dist}$  on the fusion process.

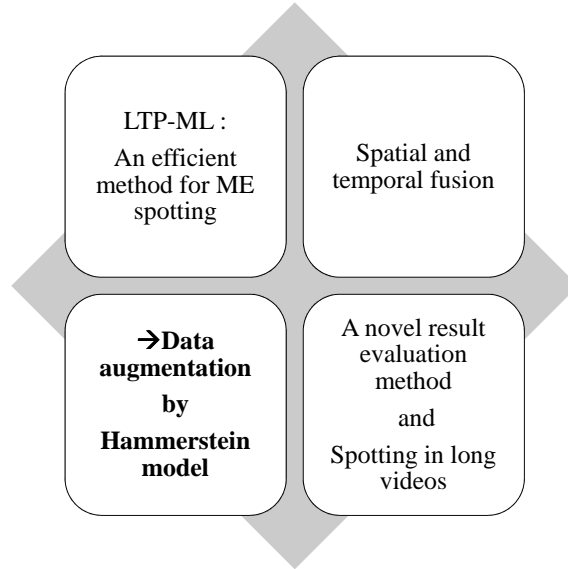
$T_{dist}$	TP	FP	FPR	Accuracy	F1-score <sub>fr</sub>
0.6	1505	2228	0.19	0.74	0.43
0.7	1486	2166	0.18	0.74	0.42
0.8	1462	2034	0.17	0.75	0.42
0.9	1359	1798	0.15	0.76	0.41
<b>1</b>	<b>1149</b>	1469	<b>0.12</b>	<b>0.77</b>	0.40

Concerning the threshold setting for normalisation coefficient ( $T_{CN}$ ), because for different ROI sequence, the movement magnitude is different, it is not reasonable to set a specific value for different samples. Hence, the average value of CN ( $\overline{CN}$ ) of each ROI sequence is set as the baseline. The threshold range is shown in Table 5.13, the larger the  $T_{CN}$  is, the more FPs there are. Yet, when  $T_{CN}$  get too small, too many qualified S-patterns are eliminated, which affects the spotting performance. The  $\overline{CN}$  is set as the  $T_{CN}$ , because it can get a good spotting result while the FP amount is acceptable.

Table 5.13: Impact of  $T_{CN}$  on the fusion process.  $\overline{CN}$  means the average value of normalization coefficient for each ROI sequence.

$T_{CN}$	TP	FP	FPR	Accuracy	F1-score <sub>fr</sub>
$0.8 \times \overline{CN}$	428	522	0.04	0.79	0.21
$0.9 \times \overline{CN}$	797	1044	0.09	0.78	0.32
<b><math>1 \times \overline{CN}</math></b>	<b>1149</b>	<b>1469</b>	<b>0.12</b>	<b>0.77</b>	<b>0.40</b>
$1.1 \times \overline{CN}$	1369	2022	0.17	0.75	0.42
$1.2 \times \overline{CN}$	1485	2332	0.20	0.73	0.42

## 5.4 Data Augmentation by Hammerstein Model for ME spotting



This section presents the third contribution of my thesis : data augmentation by Hammerstein model. The whole process: LTP-SpFS (our method with Hammerstein model) is firstly evaluated, and compared with the SOA LBP- $\chi^2$  and with LTP-ML method. In addition, the S-pattern synthesizing by Hammerstein model is also compared with a simple GAN. The spotting result shows that Hammerstein model is more appropriate for the data augmentation of micro-expression spotting. Then, we investigate the impact of LTP filtering and S-pattern synthesizing on the entire process. The impact of parameters of these two sub-steps are also analyzed separately. Finally, we study the distribution of  $(\alpha, \beta)$  in the linear module. The comparison of different distribution models of  $(\alpha, \beta)$  : normal distribution and Poisson distribution shows that the model choice has few influences on the final spotting result.

### 5.4.1 Improvement of Spotting Performance by Data Augmentation Using Hammerstein Model

In order to increase the size of S-pattern dataset for training, the LTP pattern selection of LTP-ML is replaced by LTP filtering and S-pattern synthesizing. Our method with Hammerstein model is called as LTP-SpFS. The results of LTP-SpSF are compared with the SOA  $(LBP-\chi^2)^+$  method ( $LBP-\chi^2$  with a supplementary merge process - see subsection 5.2.1) and with LTP-ML. Table 5.14 shows the spotting result. The spotting results are

Table 5.14: Result evaluation of LTP-SpFS and comparison with state-of-art method.  $(LBP-\chi^2)^+$  method represents the  $LBP-\chi^2$  with a supplementary merge process.  $F1\text{-score}_{fr}$  means F1-score of an evaluation per frame;  $F1\text{-score}_I$  means the metric proposed by MEGC (F1-score of an evaluation per interval). Our proposed LTP-SpFS method improves the spotting performance of LTP-ML and outperforms the SOA  $LBP-\chi^2$  method in terms of F1-score (both metrics).

	Database	CASME I-A	CASME I-B	CASME II
F1-score <sub>fr</sub>	$(LBP-\chi^2)^+$	0.20	0.24	0.35
	LTP-ML	0.40	0.38	0.59
	LTP-SpFS	<b>0.44</b>	<b>0.43</b>	<b>0.61</b>
F1-score <sub>I</sub>	$(LBP-\chi^2)^+$	0.09	0.06	0.12
	LTP-ML	0.26	0.23	0.42
	LTP-SpFS	<b>0.31</b>	<b>0.28</b>	<b>0.47</b>

evaluated by two metrics: one is F1-score per frame, the other one is F1-score per interval which is proposed by MEGC [64]. LTP-SpSF outperforms these two methods in both databases thanks to the extension of the training dataset by employing Hammerstein model.

### 5.4.2 Hammerstein Model is More Appropriate than GAN for ME Spotting

Hammerstein model is a traditional simulation model, which can be used for the modeling of isometric muscle dynamics. A recent method for generation, largely used in data augmentation, is GAN [33] (Generative Adversarial Network). In this subsection; we compare both method to prove the superiority of our proposed S-pattern synthesizing.

We utilize a basic GAN network to synthesize S-patterns. Since for one database, there are just around 100 S-pattern samples for training, a shallow network is utilized for the generation. The structure of our GAN is illustrated in Figure 5-9.

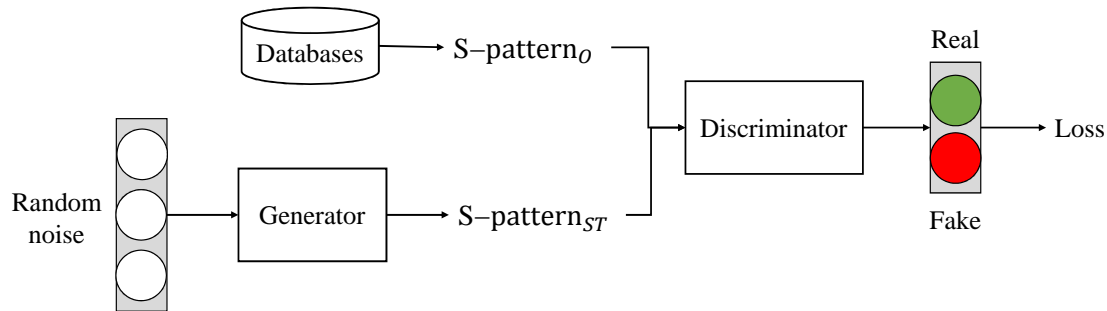
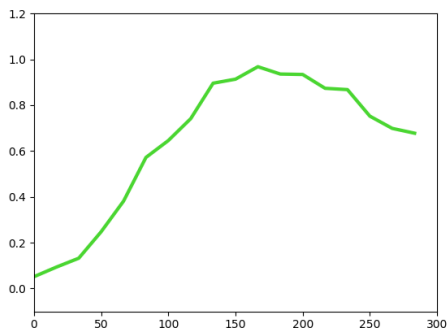


Figure 5-9: GAN structure for S-pattern synthesizing. Ransom noise is used as the input, the generator is trained to generate S-patterns, and the discriminator will compare the synthesized pattern with S-patterns from databases. Generator and discriminator are two shallow convolution networks.

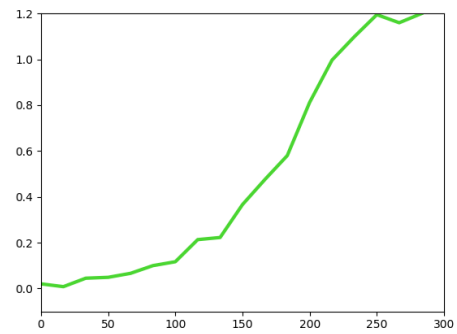
Figure 5-10 shows four samples of S-pattern generated by GAN. GAN synthesizes the S-patterns more randomly than Hammerstein model, the process generates LTP patterns instead of reliable S-patterns. Yet, all generated curves are labeled as S-patterns and then feed into the training process. It imports noise for training the classifier. Table 5.15 lists the best spotting result for S-pattern synthesized by GAN and Hammerstein model. The F1-score<sub>fr</sub> (F1-score per frame) values for GAN in both databases are smaller than these of Hammerstein model. For the further analysis, the SVM classifier is trained in two conditions:

- Situation 1: training set which contains both original and synthesized S-patterns (S-patterns<sub>O</sub>+S-patterns<sub>ST</sub>)
- Situation 2: training set which contains only synthesized S-patterns. The original S-patterns are not involved in the training of SVM classifier.

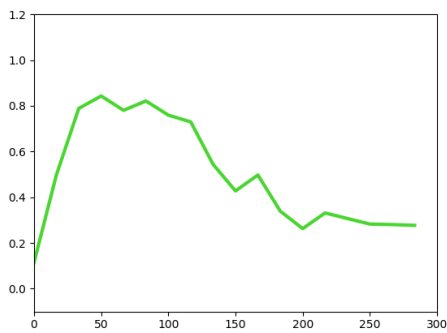
The spotting result with GAN in situation B is worse than that in situation A. That is because GAN generates S-patterns without restricted conditions, and some synthesized curve shape could not be treated as qualified S-patterns. The SVM classifier trained only



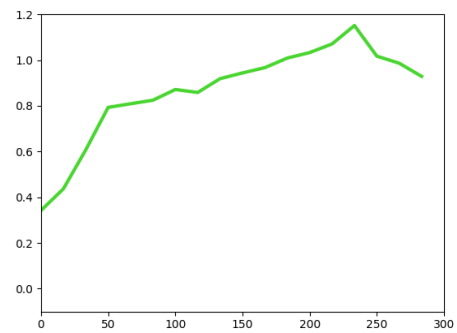
(a)



(b)



(c)



(d)

Figure 5-10: Synthesized S-pattern samples for CASME I-section A by GAN. Only 5-10a represents well the S-pattern. The other three LTPs can not be treated as reliable S-patterns. 5-10b and 5-10c show the movement with onset too long or too short. 5-10d is a on-going movement which has longer duration than micro-expression.

with synthesized patterns by GAN is not able to spot ME accurately. Otherwise, the S-patterns synthesized by Hammerstein model are more similar to original S-patterns (S-patterns<sub>O</sub>), the performance is barely influenced whether the classifier is trained with or without S-patterns<sub>O</sub>.

Table 5.15: S-patterns synthesized by Hammerstein model outperform that generated by GAN for micro-expression spotting.

		GAN	Hammerstein model
CASME I-A	With S-pattern <sub>O</sub>	0.40	<b>0.44</b>
	Without S-pattern <sub>O</sub>	0.22	<b>0.44</b>
CASME I-B	With S-pattern <sub>O</sub>	0.39	<b>0.43</b>
	Without S-pattern <sub>O</sub>	0.05	<b>0.42</b>
CASME II	With S-pattern <sub>O</sub>	0.57	<b>0.61</b>
	Without S-pattern <sub>O</sub>	0.03	<b>0.60</b>

In conclusion, the data augmentation by a simple GAN model is less performant than S-pattern synthesizing by Hammerstein model. The performance of GAN might be improved when there are more data for training. In the case that there is just a few features, a synthetic model is needed. Even GAN is popular, it does not mean it is suitable in our case.

### 5.4.3 Analysis Combining LTP Filtering and S-pattern Synthesizing

Our method with Hammerstein model contains two processes: LTP filtering (LTP-SpF) and S-pattern synthesizing (LTP-SpS). In this part, we compare the impact of these two processes separately. The baseline result is LTP-ML. The spotting results and their corresponding S-pattern amounts are listed in Table 5.16. The combination of LTP filtering and S-pattern synthesizing, i.e. LTP-SpFS performs better than LTP-SpF and LTP-SpS respectively. This is due to more reliable S-patterns. Moreover, S-pattern synthesizing (LTP-SpS) improves the ME spotting performance compared with LTP-ML by synthesizing more reliable S-patterns.

Yet, the spotting performance of LTP-SpF (LTP filtering) varies depending on the databases. It performs slightly better than LTP-ML in CASME I but not in CASME II. Indeed, LTP filtering conserves more unqualified S-pattern than LTP pattern selection. It is possible that the SVM model for CASME II is trained with more wrongly-labeled S-



Table 5.16: ME spotting result in terms of F1-score $_{fr}$  with data augmentation by Hammerstein model. LTP-ML represents our method without Hammerstein model (HM); LTP-SpF is our method with HM but only LTP filtering; LTP-SpS is our method with HM but only S-pattern synthesizing; LTP-SpFS represents the whole process with Hammerstein model (LTP filtering + S-pattern synthesizing). The spotting results for SpF and SpS are better than LTP-ML in CASME I because the size of S-pattern dataset for training stage is increased. In addition, the combination of these two steps improves the spotting performance due to a larger data volume of S-pattern.

	CASME I-A	CASME I-B	CASME II
LTP-ML	0.40	0.38	0.59
LTP-SpF	0.40	0.39	0.56
LTP-SpS	0.42	0.42	0.60
LTP-SpFS	<b>0.44</b>	<b>0.43</b>	<b>0.61</b>

(a) ME spotting result with data augmentation.

	CASME I-A	CASME I-B	CASME II
LTP-ML	78	500	2567
LTP-SpF	2776	4308	13569
LTP-SpS	1170	2500	16540
LTP-SpFS	13880	21402	19589

(b) Data amount of LTP training set after the four different processes.

patterns, and the spotting performance is affected.

We can also notice that for the whole process LTP-SpFS (our method with LTP filtering and S-pattern synthesizing), the improvement for CASME II is not as good as that for CASME I. To explain this observation, we analyze the spotting result of LTP-SpFS in CASME II for different generation times  $n$ , as shown in Figure 5-11. The performance begins to decline when the ratio of S-patterns are bigger than 0.3. Lots of unreliable S-patterns are still conserved after the LTP filtering process. Then, the synthesizing process increases their amount in the training dataset. Therefore, the spotting result is influenced by these wrongly-labeled S-patterns.

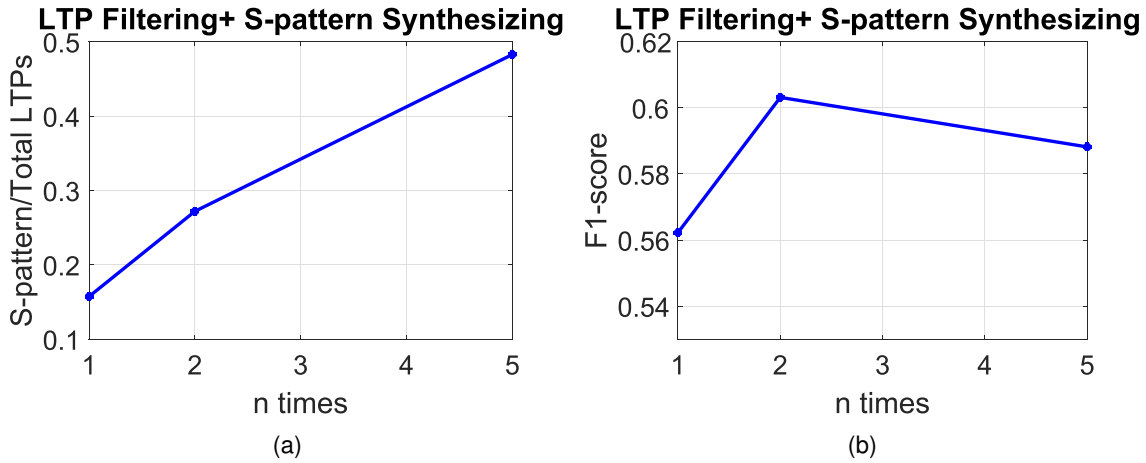


Figure 5-11: Result Evaluation according to the generation times  $n$  for the combination of LTP filtering and S-pattern synthesizing on CASME II.  $T_E$  is set to 0.25 for LTP filtering. 5-11a shows the ratio between S-pattern and the total quantity of LTPs for training. And 5-11b illustrates the F1-score for ME spotting. The parameter  $n$  for S-pattern synthesizing stops at 5 for CASME II, because the amount of S-patterns in training dataset is large enough and the F1-score have already begun to decline. By comparing these two figures, when the proportion of S-patterns is around 0.3, the spotting method performs best. Otherwise, the data augmentation process also synthesizes more wrongly-labeled S-patterns. It would import extra noise into the training process and then influence the spotting performance.

#### 5.4.4 Impact of Threshold Value for LTP Filtering

One main parameter for LTP filtering is the threshold of estimation error of Hammerstein model ( $T_E$ ). Hence, we analyze the influence of  $T_E$  on the amount of original S-pattern after LTP filtering ( $S\text{-patterns}_{OF}$ ) and also on the spotting performance. The process filters

LTP patterns by Hammerstein model with the threshold  $T_E$ . When the threshold value gets larger, the conserved S-pattern amount becomes larger. As shown in Figure 5-12, the spotting performance is improved along with the increasing of the S-pattern amount. However, more noise is brought to the classification system along with the data volume augmentation since more different pattern types are conserved. Thus, F1-score $_{fr}$  (F1-score per frame) begin to decrease when the false positives are significant. In each database, in order to find the optimal value for the spotting performance, all the samples are utilized to learn the threshold of LTP filtering .

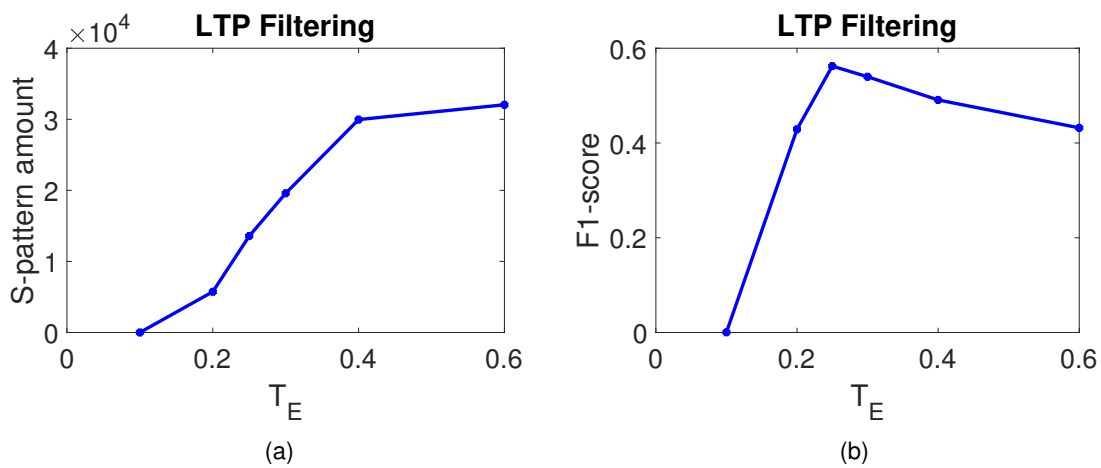


Figure 5-12: Result Evaluation according to the error threshold value for LTP filtering process. 5-12a shows the data augmentation of the S-pattern in the training dataset. More S-patterns are conserved while the threshold is larger. The increasing trend stops when the threshold can not filter any patterns. Unlike the curve for S-pattern amount, in 5-12b the curve of F1-score $_{fr}$  increases at the beginning when there are more samples for training, then it starts to decline as the filtering process conserves too many wrongly-labeled S-patterns.

### 5.4.5 Impact of $n$ for S-pattern Synthesizing

The generation multiple  $n$  is an essential input of S-pattern synthesizing process.  $n$  defines the amount of synthesized S-patterns: for one original S-pattern, the process would generate  $n$  synthesized S-patterns. Hence, we investigate the impact of  $n$  on the spotting result with this process. The amount of S-patterns in the training set after S-pattern synthesizing is  $n + 1$  times of that before the process. As illustrated in Figure 5-13, the bigger the

S-pattern amount is, the higher the F1-score<sub>fr</sub> (F1-score per frame) is. When the total S-pattern amount is 5 times of S-pattern<sub>OF</sub> amount, the slope stabilizes as the feature amount reaches saturation for the training. Thus,  $n$  is set to 5 for our experiments.

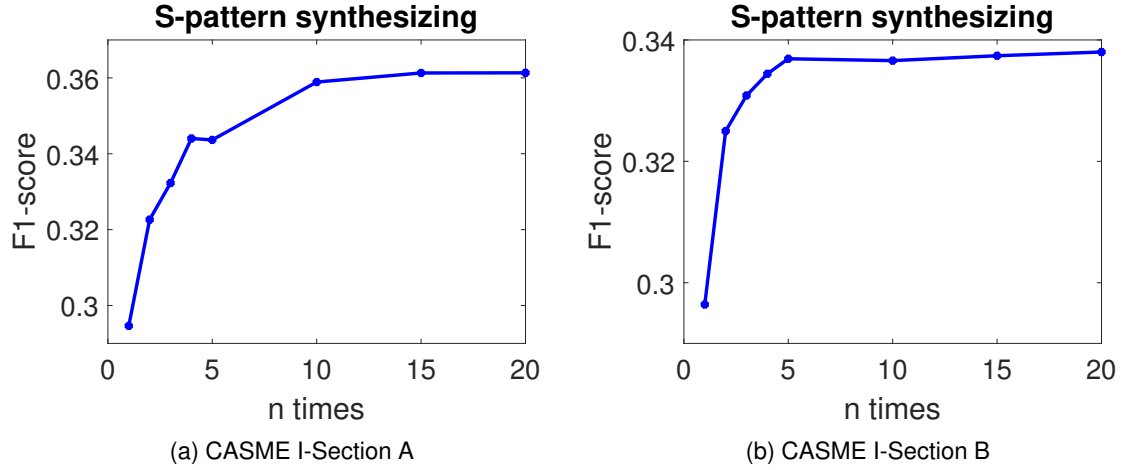


Figure 5-13: Improvement of F1-score<sub>fr</sub> by increasing the training set volume with synthesized S-patterns from Hammerstein model. The result is evaluated depending on the S-pattern amount. x axis means the generation multiple  $n$  of S-pattern<sub>O</sub>. Y axis is the F1-score. As the quantity of S-pattern increases, the F1-score<sub>fr</sub> value becomes higher than that of S-pattern<sub>OF</sub>.

#### 5.4.6 S-pattern Synthesizing by Poisson Distribution of $\alpha$ and $\beta$

To generate more synthesized S-patterns, the distribution of  $\alpha$  and  $\beta$  for qualified original S-patterns (S-pattern<sub>O</sub>) is analyzed. As illustrated in Figure 5-14, the distribution fits the Poisson distribution.

The Poisson distribution can be represented as :

$$p(k) = \frac{\lambda^k}{k!} e^{-\lambda} \quad (5.9)$$

where  $\lambda$  is the mean value of the distribution. By setting  $\lambda$  as  $\bar{\alpha}$  and  $\bar{\beta}$  separately,  $(\alpha_i, \beta_i)$  set for S-pattern synthesizing can be obtained as:

$$\alpha_i = \text{random}(\text{'Poisson'}, \bar{\alpha}), \beta_i = \text{random}(\text{'Poisson'}, \bar{\beta}) \quad (5.10)$$

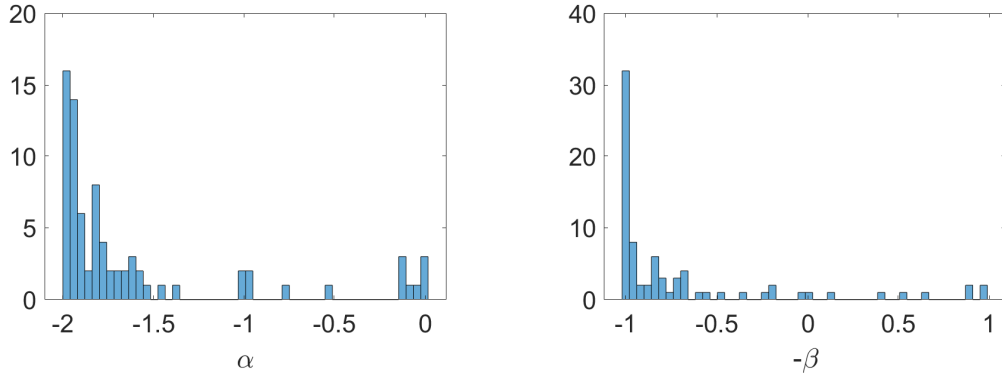


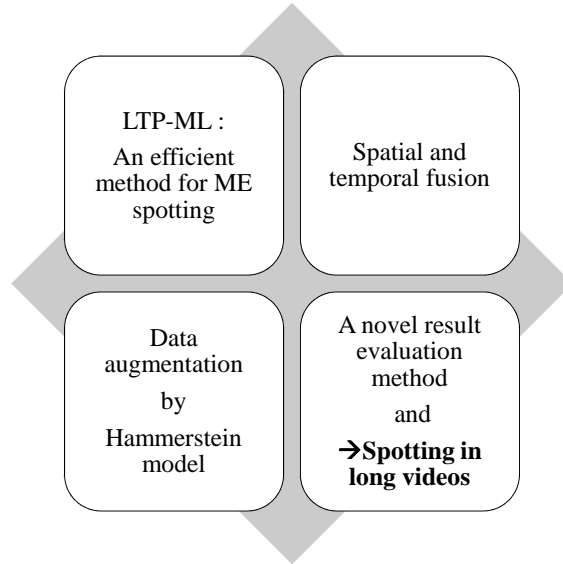
Figure 5-14: Histogram of  $\alpha$  and  $\beta$  of the linear model for S-pattern

S-patterns are synthesized based on normal distribution and Poisson distribution respectively. Table 5.17 shows the comparison result. When the generation times is set to 1, Poisson distribution has a better result than the normal distribution, because Poisson distribution is more similar to the  $\alpha$  and  $\beta$  distribution, and it is able to generate more qualified S-patterns. But the difference of S-patterns (S-patterns<sub>ST</sub>) synthesized by these two distributions are not significant. When the generations times increase, they obtain the similar spotting results. Under this condition, the sample amount for training is more important for improving the spotting performance.

Table 5.17: Spotting performance comparison between S-pattern synthesizing by normal distribution and Poisson distribution.

Generation times	1	2	3	4	5
Normal Distribution	0.37	0.39	0.41	0.42	0.42
Poisson Distribution	<b>0.39</b>	<b>0.40</b>	0.42	0.42	0.42

## 5.5 Spotting Micro-Expression in Long Videos



This section shows one part of the fourth contribution of my thesis, which addresses the challenge of spotting ME on long videos sequences. It uses two most recent databases, i.e. SAMM and CAS(ME)<sup>2</sup>. Indeed, the experiments in this section is related to our contributions to the micro-expression spotting task of MEGC2019 (Micro-expression Grand Challenge) [64]. We firstly proposed the baseline result by the SOA LBP- $\chi^2$ -distance method (LBP- $\chi^2$ ) for the challenge. Then we perform our method without Hammerstein model (LTP-ML) to compare the spotting performance. We demonstrate that our proposed method is better than the SOA method: LBP- $\chi^2$ -distance in spotting MEs.

As introduced in subsection 5.1.1, SAMM and CAS(ME)<sup>2</sup> have different frame rates per second (FPS) and facial resolution. Hence, the lengths of sliding window  $W_{video}$ , the overlap size, the interval length of  $W_{ROI}$  and the ROIs size are different for these two databases. Table 5.18 lists the experimental parameters.

For CAS(ME)<sup>2</sup> database, there are 97 videos, but only 32 videos contain micro-expressions. Thus, different results are given under two conditions: one is only considering 32 videos which have ME (CAS(ME)<sub>ME</sub><sup>2</sup>), another one is to include the entire database (all 97 videos) (CAS(ME)<sup>2</sup>).

Since the raw videos in SAMM database are too big to download (700GB, 224 videos), only 79 videos were provided for the challenge. In this work, we report the results based

Table 5.18: Parameter configuration for SAMM and CAS(ME)<sup>2</sup> depending on FPS and faical resolution.  $L_{window}$  is the length of sliding window  $W_{video}$ ,  $L_{overlap}$  is the overlap size between sliding windows,  $L_{interval}$  is the interval length of  $W_{ROI}$ . The facial resolution given in the table is the average value among the entire database.

Database	FPS	Facial Resolution	$L_{window}$	$L_{overlap}$	$L_{interval}$	$size_{ROI}$
SAMM	200	400×400	200	60	60	15
CAS(ME) <sup>2</sup>	30	200×240	30	9	9	10

on these two versions of SAMM database: one is the cropped videos (only facial regions are conserved, SAMM<sub>ME</sub><sup>c</sup>) provided by the authors using the method in [23], and the other one is the videos with full frame (original video without any pre-processing) (SAMM<sub>ME</sub><sup>f</sup>). The spotting process is performed only on the downloaded databases.

### Experiments Results of LBP- $\chi^2$ -distance (LBP- $\chi^2$ ) Method

The baseline method is LBP- $\chi^2$ -distance (LBP- $\chi^2$ ) method. The spotting result is listed in Table 5.19. For CAS(ME)<sub>ME</sub><sup>2</sup>, when the threshold for peak selection is set to 0.15, we can get the best result for LBP- $\chi^2$  method, the *FI-score* is 0.0111. Meanwhile, the highest *FI-score* of SAMM<sub>ME</sub><sup>c</sup> is 0.0055 when the threshold is set to 0.05.

### Experiments Results of LTP-ML Method

After performing the LTP-ML method on these two databases, the spotting results for the whole database are listed in Table 5.19. The *FI-score* for (SAMM<sub>ME</sub><sup>c</sup>) and CAS(ME)<sub>ME</sub><sup>2</sup> are 0.0316 and 0.0179 respectively. LTP-ML performs better in SAMM<sub>ME</sub><sup>c</sup> than SAMM<sub>ME</sub><sup>f</sup>, since the cropped-face process has already aligned the face region in the video, and reduced the influence of irrelevant movements. Concerning the spotting result of CAS(ME)<sub>ME</sub><sup>2</sup>, there are more *FPs* because the video in this database which has no ME may contain macro-expressions.

**Compared with LBP- $\chi^2$  method, LTP-ML is more accurate. LTP-ML method is capable of spotting the subtle movements based on the patterns which represented the temporal pattern variation of ME.** Yet, the value of *FI-score* is low because of the large amounts of *FP*. Both databases contain noises and irrelevant facial movements, especially

Table 5.19: Micro-expression spotting result in long videos.  $SAMM_{ME}^c$  represents the SAMM cropped-face videos contain ME,  $SAMM_{ME}^f$  are the ME videos with full frame,  $CAS(ME)_{ME}^2$  means all the videos in this sub-dataset of  $CAS(ME)^2$  have ME sequences.

Method	LBP- $\chi^2$		
	$SAMM_{ME}^c$	$CAS(ME)_{ME}^2$	$CAS(ME)^2$
nb_vid	79	32	97
TP	12	10	10
FP	4172	1729	5435
FN	147	47	47
Precision	0.0028	0.0057	0.0018
Recall	0.0755	0.1754	0.1754
F1-score	0.0055	0.0111	0.0035

(a) Baseline method

Method	LTP-ML			
	$SAMM_{ME}^c$	$SAMM_{ME}^f$	$CAS(ME)_{ME}^2$	$CAS(ME)^2$
nb_vid	79	79	32	97
TP	34	47	16	16
FP	1958	3891	1711	5742
FN	125	112	41	41
Precision	0.0171	0.0043	0.0093	0.0028
Recall	0.2138	0.2956	0.2807	0.2807
F1-score	<b>0.0316</b>	0.0229	<b>0.0179</b>	<b>0.0055</b>

(b) Our proposed method



for CAS(ME)<sup>2</sup>, it is not easy to separate macro-expressions from micro-expressions based on 30fps videos. The ability of distinguishing ME from other movements still need to be enhanced.

In conclusion, whilst our method was able to produce a reasonable amount of *T*Ps, there are still a huge challenge lays ahead due to the large amount of *F*Ps. Further research should focus on enhancing the ability of distinguishing ME from other facial movements to reduce *F*Ps.

## 5.6 Conclusion

In this chapter, we demonstrate our proposed LTP pattern is relevant for micro-expression spotting. In addition, our method is more performant than the SOA LBP- $\chi^2$  method in both short video and long video databases. Furthermore, the LTP filtering and data augmentation by Hammerstein model help to improve the micro-expression spotting performance.

# Chapter 6

## Conclusion and Perspective

### 6.1 Conclusion

The Micro-expressions (MEs) are very important nonverbal communication clues. However, due to their local and short nature, spotting them is challenging. We showed in the state of arts that there are few but increasing articles on micro-expression analysis. Meantime, the spotting accuracy is low. Yet, the research begins to attract more attentions, and the community organized the first micro-expression spotting challenge in MEGC2019 (Micro-Expression Grand Challenge).

In this thesis, we addressed this problem by using a dedicated local and temporal pattern (LTP) of facial movement. This pattern has a specific shape (S-pattern) when ME are displayed. Thus, by using a classical classification algorithm (SVM), MEs are distinguished from other facial movements. We also proposed a global final fusion analysis on the whole face to improve the distinction between ME (local) and head (global) movements.

The automatic micro-expression analysis is restricted by the small size of database. There are only 6 public spontaneous micro-expression database, and the average amount of ME sequence samples for these databases is 152. A performant classifier requires more samples for training. On the other side, there is no agreement on the result evaluation methods and metrics. Each paper has its own metric which is adapted to its proposed method. Yet, the inconsistency influences the comparison between different spotting methods.

Concerning our method, the learning of S-patterns is also limited by the small number

of ME databases and the low volume of ME samples. Hammerstein models (HMs) are known to be a good approximation of muscle movements. By approximating each S-pattern with a HM, we have both filtered outliers and generate new similar S-patterns. By this way, we performed a data augmentation for S-pattern training dataset and improved the ability to differentiate micro-expressions from other facial movements.

Besides, we participated to the micro-expression spotting challenge (MEGC2019). Through an international cooperation, we proposed a novel result evaluation method per interval to evaluate the spotting performance. Meanwhile, we applied our method to spotting micro-expression in long videos and provided the baseline result for the challenge.

The spotting results, performed on CASMEI and CASMEII, SAMM and CAS(ME)<sup>2</sup>, showed that our proposed LTP outperforms the most popular spotting method: LBP- $\chi^2$ -distance in terms of F1-score. The experimental analysis was according to our contributions. Firstly, the relevancy of our proposed feature (local temporal pattern) has been proved by the comparison with another commonly used feature (LBP-TOP). The spotting per emotion verified the assumption: the S-patterns are similar for all kinds of emotions. Secondly, adding the fusion process helped enhance the ability of distinguish the micro-expressions and other movements. Thirdly, data augmentation with Hammerstein model improved even more the spotting performance. It is because that LTP filtering conserves more reliable S-patterns and S-pattern synthesizing largely increases the size of training dataset. Except the global analysis on the contributions, the impact of several parameters in the process on the spotting performance were analyzed to identify the optimal configuration of experiment.

Therefore, our contributions are highlighted as below:

- A novel relevant feature for micro-expression spotting: local temporal patterns (LTPs);
- A late spatial and temporal fusion, which helps to enhance the ability of distinguishing micro-expressions from other facial movements;
- LTP filtering and Data augmentation by Hammerstein model;
- The first micro-expression spotting challenge : a novel result evaluation method per

interval to evaluate the spotting performance, and spotting micro-expression in long videos by baseline method and our proposed method.

## 6.2 Perspectives

### 6.2.1 Perspective of Our Method

Concerning the perspective of our method, we discuss the following three points:

**1. Reducing false positives:** The micro-expression spotting ability is still weak due to the large amount of irrelevant movement and noise. In order to reduce the false positives causing by other facial movement or environment change such as eye blinks and lighting variation, further researches should focus on enhancing the ability of distinguishing ME from other facial movements, including the implementation of deep learning approaches when we have enough data.

**2. Micro-expression spot-and-recognize schema:** As long as the micro-expression is spotted in the video, the spotted clips can be used for micro-expression recognition. In our proposed method, the S-patterns are recognized from local regions, the combination of different ROI indicates the different action units, i.e. different emotion types. Therefore, the local position of spotted S-patterns can be used as a feature for micro-expression recognition.

**3. Generalization of spotting method:** We emphasize on the further improvement for our method:

- First of all, the cross-database validation of micro-expression spotting is expected. The classifier of our method is trained and tested in the same database by leave-one-subject-out-cross-validation (LOSubOCV). The further experiments can be performed in these two situations: 1). train the classifier in one database and then perform the test in another database; 2). treat multiple database as one database and performed the machine learning method by LOSubOCV.
- Secondly, the parameters in data augmentation method with Hammerstein model is specific for each database. It is worth to generalize the parameters for different

databases to develop the common S-patterns for micro-expressions. In this case, even for a database without annotations, micro-expressions can be spotted thanks to the common S-pattern.

### **6.2.2 Data Augmentation for Micro-Expression Spotting**

Since the micro-expression analysis is restricted by the amount of micro-expression samples, the data augmentation is necessary to improve the performance.

- More micro-expression databases are expected. Chinese Academy of Science has built a platform for micro-expression annotation, more micro-expression sequences may be available in the future. In addition, some macro-expression frames with small intensity can be utilized as micro-expression samples.
- Beside creating more database, synthesizing either micro-expression features (as we have done) but also video sequences is an option for data augmentation. More generation method can be exploited to synthesize reliable relevant micro-expression features.

### **6.2.3 Consistency of Metrics**

Concerning the consistency of metrics, as the machine learning is the trend for the further research of micro-expression analysis, the common metrics of machine learning method: precision, recall and F1-score are recommended. In addition, spotting micro-expression interval seems promising as it gives more samples to study the spotted movement. Yet, to reach an agreement on the result evaluation method and metrics, it requires more research and more experiments to verify which metric is more appropriate for the applications of micro-expression spotting.

### **6.2.4 Micro-Expression Spotting Applications**

We look forward to the applications for micro-expression spotting in real world.

- More research should focus on spotting micro-expression in long videos or in in-the-wild video samples (e.g. MEVIEW database [52]). Spotting micro-expression in these kinds of situation is more challenging than that in short videos samples with spontaneous micro-expression. For instance: differentiating micro-and macro-expressions, variation of lightness situation, masked face, large head movement etc.
- The micro-expression spotting can target to an applicative situation. For example, the technique can spot the symptoms when the patient in coma is going to wake up. However, it requires specific databases and annotations which are adapted to this application instead of emotion labels.
- Micro-expression spotting in real time is also expected. The task requires not only a strong ability of differentiating micro-expression and other movements, but also the rapid computation capacity.
- Fusion of macro- and micro-expression spotting could be interesting for application concerning emotion analysis.



# Glossary

## Abbreviation of Definitions

AE: Auto-Encoder

AU: Action Unit

CN: Normalisation Coefficient

F1-score<sub>*f<sub>r</sub>*</sub>: F1-score per frame

F1-score<sub>*l*</sub>: F1-score per interval

FACS: Facial Action Coding System

FN: False Negative

FP: False Positive

FPR: False Positive Rate

GAN: Generative Adversarial Network

HM: Hammerstein model

LBP: Local Binary Pattern

LBP- $\chi^2$ : LBP- $\chi^2$  distance method

LBP-TOP: Local Binary Pattern on three orthogonal planes: xy, xt and yt

LC: Local Classification

LoSubOCV: Leave-One-Subject-Out-Cross-Validation

LQ: Local Qualification

LTP: Local Temporal Pattern

LTP-ML: Micro-expression spotting by local temporal pattern without Hammerstein model

LTP-SpF: LTP filtering

LTP-SpFS: Micro-expression spotting by local temporal pattern with LTP filtering and S-



pattern synthesizing  
 LTP-SpS: S-pattern Synthesizing  
 ME: Micro-Expression  
 MESR: Micro-Expression Spotting and Recognition  
 MP: Merge Process  
 PCA: Principal Component Analysis  
 ROI: Region of Interest  
 SF: Spatial Fusion  
 SOA: State-of-atrs  
 S-pattern: Local temporal pattern when micro-expression occurs  
 STF: Spatial and Temporal Fusion  
 SVM: Support Vector Machine  
 TP: True Positive  
 TPR: True Positive Rate

### **Abbreviation of Variables and Parameters**

$(\alpha, \beta)$ : Parameters in the linear module of Hammerstein model.  
 $E_H$ : Estimation error of Hammerstein model  
 $K + 1$ : Number of frames during the average micro-expression duration, 300ms  
 $p$ : Parameters in non-linear module of Hammerstein model  
 $J$ : Amount of chosen ROIs  
 $T_{CN}$ : Threshold CN value for local qualification  
 $T_{dist}$ : Threshold distance value for LTP pattern selection and local qualification  
 $S\text{-pattern}_O$ : Original S-pattern after label annotation and AU selection (Figure 3-9).  
 $S\text{-pattern}_{OF}$ : Conserved  $S\text{-pattern}_O$  after LTP filtering.  
 $S\text{-pattern}_{OS}$ : Conserved  $S\text{-pattern}_O$  after LTP pattern selection (Figure 3-9).  
 $S\text{-pattern}_{ST}$ : Synthesized S-pattern by Hammerstein model.  
 $T_E$ : Threshold of estimation error for LTP filtering.  
 $W_{ROI}$ : Sliding window per frame with length  $K + 1$  for LTP computation.

# Publication

## International Journal

WEBER Raphaël, **Li Jingting**, SOLADIE, Catherine, et SEGUIER, Renaud. (2019) A Survey on Databases of Facial Macro-expression and Micro-expression. In: Bechmann D. et al. (eds) Computer Vision, Imaging and Computer Graphics Theory and Applications. VISIGRAPP 2018. Communications in Computer and Information Science, vol 997. Springer, Cham

## International Conference

**LI, Jingting**, SOLADIE, Catherine, SEGUIER, Renaud, Wang, Su-Jing, et Yap, Moi Hoon. Spotting Micro-Expressions on Long Videos Sequences. In : 2019 14th IEEE International Conference on Automatic Face & Gesture Recognition (FG 2019). IEEE, 2019. p. 1-5.

SEE, John, YAP, Moi Hoon, **LI, Jingting**, Hong, Xiaopeng, et Wang, Su-Jing. MEGC 2019 – The Second Facial Micro-Expressions Grand Challenge. In : 2019 14th IEEE International Conference on Automatic Face & Gesture Recognition (FG 2019). IEEE, 2019. p. 1-5.

**LI, Jingting**, SOLADIE, Catherine, et SEGUIER, Renaud. A Survey on Databases for Facial Micro-expression Analysis. VISIGRAPP(5: VISAPP).2019

**LI, Jingting**, SOLADIE, Catherine, et SEGUIER, Renaud. LTP-ML: Micro-Expression Detection by Recognition of Local Temporal Pattern of Facial Movements. In : Automatic Face & Gesture Recognition (FG 2018), 2018 13th IEEE International Conference on. IEEE, 2018. p. 634-641

### **National Conference**

**LI, Jingting**, SOLADIE, Catherine, et SEGUIER, Renaud. Détection de Micro-expressions par Reconnaissance de Motif Local Temporel de Mouvements Faciaux. apex, 2017, vol. 1, p. 1.5. (Reconnaissance des Formes, Image, Apprentissage et Perception (RFIAP), 2018)








### **Community activity**











Oral presentation at «Réunion du GdR ISIS : Journée Action, Visage, geste, action et comportement» : A survey on Automatic Facial Micro-expression Spotting: Databases, Metrics and Methods



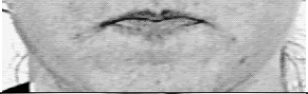






Organisation of The Third Facial Micro-Expressions Grand Challenge (MEGC2020)



# Appendix A

## FACS - Facial Action Coding System[1]

AU	Description	Facial muscle	Example image
1	Inner Brow Raiser	<i>Frontalis, pars medialis</i>	
2	Outer Brow Raiser	<i>Frontalis, pars lateralis</i>	
4	Brow Lowerer	<i>Corrugator supercilii, Depressor supercilii</i>	
5	Upper Lid Raiser	<i>Levator palpebrae superioris</i>	
6	Cheek Raiser	<i>Orbicularis oculi, pars orbitalis</i>	
7	Lid Tightener	<i>Orbicularis oculi, pars palpebralis</i>	
9	Nose Wrinkler	<i>Levator labii superioris alaequae nasi</i>	

10	Upper Lip Raiser	<i>Levator labii superioris</i>	
11	Nasolabial Deepener	<i>Zygomaticus minor</i>	
12	Lip Corner Puller	<i>Zygomaticus major</i>	
13	Cheek Puffer	<i>Levator anguli oris (a.k.a. Caninus)</i>	
14	Dimpler	<i>Buccinator</i>	
15	Lip Corner Depressor	<i>Depressor anguli oris (a.k.a. Triangularis)</i>	
16	Lower Lip Depressor	<i>Depressor labii inferioris</i>	
17	Chin Raiser	<i>Mentalis</i>	
18	Lip Puckerer	<i>Incisivii labii superioris and Incisivii labii inferioris</i>	
20	Lip stretcher	<i>Risorius w/ platysma</i>	

22	Lip Funneler	<i>Orbicularis oris</i>	
23	Lip Tightener	<i>Orbicularis oris</i>	
24	Lip Pressor	<i>Orbicularis oris</i>	
25	Lips part	<i>Depressor labii inferioris or relaxation of Mentalis, or Orbicularis oris</i>	
26	Jaw Drop	<i>Masseter, relaxed Temporalis and internal Pterygoid</i>	
27	Mouth Stretch	<i>Pterygoids, Digastric</i>	
28	Lip Suck	<i>Orbicularis oris</i>	
41	Lid droop	<i>Relaxation of Levator palpebrae superioris</i>	
42	Slit	<i>Orbicularis oculi</i>	

43	Eyes Closed	<p><i>Relaxation of Levator palpebrae superioris;</i>  <i>Orbicularis oculi, pars palpebralis</i></p>	
44	Squint	<p><i>Orbicularis oculi, pars palpebralis</i></p>	
45	Blink	<p><i>Relaxation of Levator palpebrae superioris;</i>  <i>Orbicularis oculi, pars palpebralis</i></p>	
46	Wink	<p><i>Relaxation of Levator palpebrae superioris;</i>  <i>Orbicularis oculi, pars palpebralis</i></p>	

## **Appendix B**

# **Summary Table for Published Micro-Expression Recognition Articles with Their Corresponding Metrics and Databases**

Table B.1 comprehensively lists the published articles of micro-expression recognition.

And all the articles are classified into their corresponding metrics and databases.

CASME II [135] and SMIC [66] are two most used databases. The number of articles using SAMM [19] and CAS(ME)<sup>2</sup> [104] are still small. This is due to that they are still new for the community. Since these two databases contain long video samples, we look forward that more research could perform their experiments on them.



Table B.1: Summary table for published micro-expression recognition articles with their corresponding metrics and databases

	CASMEI	CASMEII	SMIC/SMIC-E	SAMM	CAS(ME) <sup>2</sup>
Accuracy	[113, 117, 118, 82, 120, 97, 114, 85, 138, 38, 140, 42, 97, 110, 133, 146, 43, 84, 49, 51, 47, 81]	[117, 121, 136, 118, 82, 120, 97, 114, 61, 73, 123, 93, 125, 59, 76, 62, 75, 80, 122, 72, 85, 95, 7, 57, 141, 5, 48, 9, 10, 14, 42, 45, 55, 70, 94, 97, 110, 133, 142, 146, 148, 8, 43, 84, 99, 22, 21, 90, 48, 48, 65, 49, 51, 47, 50, 150, 145, 69, 32, 71, 81]	[121, 82, 114, 61, 73, 123, 125, 76, 62, 75, 80, 72, 36, 85, 138, 46, 140, 14, 42, 45, 133, 146, 8, 43, 84, 100, 66, 65, 49, 51, 47, 50, 150, 145, 32, 71, 81]	[72, 99, 19, 21, 90]	[104, 84]
Confusion matrix	[118, 82, 97, 147, 97, 110, 133, 146, 84, 49, 51, 47, 81]	[118, 82, 97, 76, 92, 80, 122, 56, 7, 147, 5, 14, 97, 110, 133, 146, 84, 99, 151, 90, 48, 49, 51, 47, 50, 149, 145, 81]	[82, 76, 92, 80, 13, 133, 146, 84, 151, 48, 49, 51, 47, 50, 149, 145, 81]	[56, 99, 90]	[84]
F1-score	[120, 42, 133, 43, 81]	[120, 61, 93, 59, 76, 92, 62, 75, 80, 56, 58, 72, 5, 14, 42, 133, 43, 99, 21, 90, 150, 81]	[61, 76, 92, 62, 75, 80, 72, 14, 42, 133, 43, 150, 81]	[56, 72, 99, 19, 21, 90]	
Recall	[140, 54, 81]	[61, 93, 59, 92, 62, 75, 58, 7, 5, 14, 21, 90, 81]	[61, 92, 62, 75, 140, 14, 54, 81]	[21, 90]	
Precision	[140, 81]	[61, 93, 59, 92, 62, 75, 58, 5, 5, 14, 81]	[61, 92, 62, 75, 140, 14, 81]		
Time	[84]	[125, 76, 75, 84, 150]	[125, 76, 75, 84, 150]		[84]
ROC	[42, 54]	[42, 21, 90]	[42, 54]	[21, 90]	[104]

# Appendix C

## Feature Extraction for Micro-Expression Spotting in Long Videos

In this appendix, the LTP computation process of chapter 3 is updated for the long video situation. The subscript of formulas in LTP feature extraction is changed, because there multiple short sequences in one long video.

After the pre-processing in subsection 3.4.1, local temporal patterns (LTPs) [63] are analyzed in the local region to distinguish micro-expression from other movements. They are extracted from 12 ROIs (Regions of Interest) respectively in each short sequence. Supposing there are  $M$  short sequences in one long video. Then in the short video  $I_m$  ( $m \leq M$ ), the  $j$ th ROI sequence is noted as  $ROI_j^m$  ( $j \leq 12$ ). The following paragraphs introduce the LTP computation process in one short ROI sequence:  $ROI_j^m$ .

### Main Local Movement Extraction by PCA

PCA is performed on the temporal axis to conserve the main distortion of the grey level texture. The first two components of each ROI frame are used to analyze the variation pattern of local movement. The process can be presented as in equation C.1.

$$\begin{bmatrix} P_1^{m,j}(x) & \cdots & P_N^{m,j}(x) \\ P_1^{m,j}(y) & \cdots & P_N^{m,j}(y) \end{bmatrix} = \Phi \times \left( \begin{bmatrix} F_1^{m,j}(1) & \cdots & F_N^{m,j}(1) \\ \vdots & & \vdots \\ F_1^{m,j}(a^2) & \cdots & F_N^{m,j}(a^2) \end{bmatrix} - \bar{I} \right) \quad (\text{C.1})$$

where  $F_n^{m,j}$  represents the pixels in one ROI frame,  $P_n^{m,j} = [P_n^{m,j}(x), P_n^{m,j}(y)]$  are the first two components of PCA,  $n$  is the frame index in this ROI sequence ( $n \leq N$ , where  $N$  is the total number of frames in this sequence). Hence, each frame in  $ROI_j^m$  can be represented by a point  $P_n^{m,j}$ .

### LTP Extraction: Distance Computation

A sliding window  $W_{ROI}$  per frame with length  $K + 1$  (300ms, the average duration of ME) is performed on  $ROI_j^m$ . The distances between the first frame and the other frames in this window are calculated. The window goes through each frame in the sequence  $ROI_j^m$ , and the distance set can be got as  $[\Delta_j^m(n, n + 1), \Delta_j^m(n, n + k), \dots, \Delta_j^m(n, n + K)]$ , as shown in

Figure C-1.

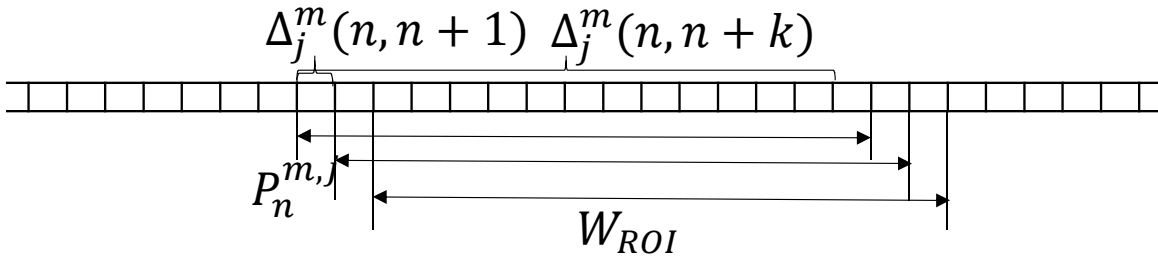


Figure C-1: Distance calculation for one ROI sequence  $ROI_j^m$  in video clip  $I_m$ .

The values of distance are then normalized for the entire  $ROI_j^m$  to avoid the influence of different movement magnitude in different videos. Hence, the feature of frame  $n$  for  $ROI_j^m$  can be represented as:  $[CN_j^m, d_j^m(n, n + 1), \dots, d_j^m(n, n + K)]$ , where  $d_j^m(n, n + k)$  is the normalized distance value and the  $CN_j^m$  is the normalization coefficient.

# Bibliography

- [1] <https://www.cs.cmu.edu/~face/facs.htm>.
- [2] <http://www.dynamixyz.com/>.
- [3] <https://ww2.mathworks.cn/help/ident/ref/systemidentification-app.html>.
- [4] <https://ww2.mathworks.cn/help/ident/ref/pwlinear.html>.
- [5] Muhammad Navid Anjum Aadit, Mehnaz Tabassum Mahin, and Shamima Nasrin Juthi. Spontaneous micro-expression recognition using optimal firefly algorithm coupled with iso-flann classification. In *Humanitarian Technology Conference (R10-HTC), 2017 IEEE Region 10*, page 714–717. IEEE, 2017.
- [6] Hervé Abdi and Lynne J. Williams. Principal component analysis. *Wiley interdisciplinary reviews: computational statistics*, 2(4):433–459, 2010.
- [7] Iyanu Pelumi Adegun and Hima B. Vadapalli. Automatic recognition of micro-expressions using local binary patterns on three orthogonal planes and extreme learning machine. In *Pattern Recognition Association of South Africa and Robotics and Mechatronics International Conference (PRASA-RobMech), 2016*, page 1–5. IEEE, 2016.
- [8] B. Allaert, I. M. Bilasco, and C. Djeraba. Advanced local motion patterns for macro and micro facial expression recognition. *arXiv preprint arXiv:1805.01951*, 2018.
- [9] Benjamin Allaert, Ioan Marius Bilasco, Chabane Djeraba, Benjamin Allaert, José Mennesson, Ioan Marius Bilasco, Chabane Djeraba, Afifa Dahmane, Slimane Larabi, and Ioan Marius Bilasco. Consistent optical flow maps for full and micro facial expression recognition. In *VISIGRAPP (5: VISAPP)*, page 235–242, 2017.
- [10] Xianye Ben, Xitong Jia, Rui Yan, Xin Zhang, and Weixiao Meng. Learning effective binary descriptors for micro-expression recognition transferred by macro-information. *Pattern Recognition Letters*, 2017.
- [11] Leonas A Bernotas, Patrick E Crago, and Howard J Chizeck. A discrete-time model of electrically stimulated muscle. *IEEE transactions on biomedical engineering*, (9):829–838, 1986.

- [12] Ray L Birdwhistell. Communication without words. *Eristics*, pages 439–444, 1968.
- [13] Diana Borza, Radu Danescu, Razvan Itu, and Adrian Darabant. High-speed video system for micro-expression detection and recognition. *Sensors*, 17(12):2913, 2017.
- [14] Diana Borza, Radu Danescu, Razvan Itu, and Adrian Darabant. High-speed video system for micro-expression detection and recognition. *Sensors*, 17(12):2913, Dec 2017.
- [15] Diana Borza, Razvan Itu, and Radu Danescu. Micro expression detection and recognition from high speed cameras using convolutional neural networks. In *VISIGRAPP (5: VISAPP)*, 2018.
- [16] Chih-Chung Chang and Chih-Jen Lin. Libsvm: a library for support vector machines. *ACM transactions on intelligent systems and technology (TIST)*, 2(3):27, 2011.
- [17] Mei-Hung Chiu, Hongming Leonard Liaw, Yuh-Ru Yu, and Chin-Cheng Chou. Facial micro-expression states as an indicator for conceptual change in students’ understanding of air pressure and boiling points. *British Journal of Educational Technology*.
- [18] Danielle Chou. *Efficacy of Hammerstein models in capturing the dynamics of isometric muscle stimulated at various frequencies*. PhD thesis, Massachusetts Institute of Technology, 2006.
- [19] A. K. Davison, C. Lansley, N. Costen, K. Tan, and M. H. Yap. Samm: A spontaneous micro-facial movement dataset. *IEEE Transactions on Affective Computing*, 9(1):116–129, Jan 2018.
- [20] Adrian Davison, Walied Merghani, Cliff Lansley, Choon-Ching Ng, and Moi Hoon Yap. Objective micro-facial movement detection using face-based regions and baseline evaluation. In *Automatic Face & Gesture Recognition (FG 2018), 2018 13th IEEE International Conference on*, pages 642–649. IEEE, 2018.
- [21] Adrian K Davison, Walied Merghani, and Moi Hoon Yap. Objective classes for micro-facial expression recognition(submitted). *Royal Society open science*.
- [22] Adrian K. Davison, Moi Hoon Yap, Nicholas Costen, Kevin Tan, Cliff Lansley, and Daniel Leightley. Micro-facial movements: an investigation on spatio-temporal descriptors. In *European conference on computer vision*, page 111–123. Springer, 2014.
- [23] Adrian K. Davison, Moi Hoon Yap, and Cliff Lansley. Micro-facial movement detection using individualised baselines and histogram-based descriptors. In *2015 IEEE International Conference on Systems, Man, and Cybernetics*, page 1864–1869. IEEE, 2015.

- [24] Carlos Duque, Olivier Alata, Rémi Emonet, Anne-Claire Legrand, and Hubert Konik. Micro-expression spotting using the riesz pyramid. In *WACV 2018*, 2018.
- [25] Paul Eckman. Emotions revealed. *St. Martin's Griffin, New York*, 2003.
- [26] Paul Ekman. Lie catching and microexpressions. *The philosophy of deception*, page 118–133, 2009.
- [27] Paul Ekman and Wallace Friesen. Facial action coding system: a technique for the measurement of facial movement. *Palo Alto: Consulting Psychologists*, 1978.
- [28] Paul Ekman and Wallace V. Friesen. Nonverbal leakage and clues to deception. *Psychiatry*, 32(1):88–106, 1969.
- [29] Paul Ekman and Wallace V Friesen. Constants across cultures in the face and emotion. *Journal of personality and social psychology*, 17(2):124, 1971.
- [30] Jennifer Endres and Anita Laidlaw. Micro-expression recognition training in medical students: a pilot study. *BMC medical education*, 9(1):47, 2009.
- [31] MG Frank, Malgorzata Herbasz, Kang Sinuk, A Keller, and Courtney Nolan. I see how you feel: Training laypeople and professionals to recognize fleeting emotions. In *The Annual Meeting of the International Communication Association. Sheraton New York, New York City*, 2009.
- [32] YS Gan and Sze-Teng Liong. Bi-directional vectors from apex in cnn for micro-expression recognition.
- [33] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in neural information processing systems*, pages 2672–2680, 2014.
- [34] Jelena Grobova, Milica Colovic, Marina Marjanovic, Angelina Njegus, Hasan Demire, and Gholamreza Anbarjafari. Automatic hidden sadness detection using micro-expressions. In *Automatic Face & Gesture Recognition (FG 2017), 2017 12th IEEE International Conference on*, pages 828–832. IEEE, 2017.
- [35] Jianzhu Guo, Shuai Zhou, Jinlin Wu, Jun Wan, Xiangyu Zhu, Zhen Lei, and Stan Z. Li. Multi-modality network with visual and geometrical information for micro emotion recognition. In *Automatic Face & Gesture Recognition (FG 2017), 2017 12th IEEE International Conference on*, page 814–819. IEEE, 2017.
- [36] Yanjun Guo, Yantao Tian, Xu Gao, and Xuange Zhang. Micro-expression recognition based on local binary patterns from three orthogonal planes and nearest neighbor method. In *Neural Networks (IJCNN), 2014 International Joint Conference on*, page 3473–3479. IEEE, 2014.
- [37] Yingchun Guo, Cuihong Xue, Yingzi Wang, and Ming Yu. Micro-expression recognition based on cbp-top feature with elm. *Optik-International Journal for Light and Electron Optics*, 126(23):4446–4451, 2015.

- [38] Yingchun Guo, Cuihong Xue, Yingzi Wang, and Ming Yu. Micro-expression recognition based on cbp-top feature with elm. *Optik-International Journal for Light and Electron Optics*, 126(23):4446–4451, 2015.
- [39] Ernest A. Haggard and Kenneth S. Isaacs. Micromomentary facial expressions as indicators of ego mechanisms in psychotherapy. In *Methods of research in psychotherapy*, page 154–165. Springer, 1966.
- [40] Yiheng Han, Bingjun Li, Yu-Kun Lai, and Yong-Jin Liu. Cfd: A collaborative feature difference method for spontaneous micro-expression spotting. In *2018 25th IEEE International Conference on Image Processing (ICIP)*, pages 1942–1946. IEEE, 2018.
- [41] Xiao-li Hao and Miao Tian. Deep belief network based on double weber local descriptor in micro-expression recognition. In *Advanced Multimedia and Ubiquitous Engineering*, pages 419–425. Springer, 2017.
- [42] S. L. Happy and Aurobinda Routray. Fuzzy histogram of optical flow orientations for micro-expression recognition. *IEEE Transactions on Affective Computing*, 2017.
- [43] S. L. Happy and Aurobinda Routray. *Recognizing Subtle Micro-facial Expressions Using Fuzzy Histogram of Optical Flow Orientations and Feature Selection Methods*, page 341–368. Springer, 2018.
- [44] Jiachi He, Jian-Fang Hu, Xi Lu, and Wei-Shi Zheng. Multi-task mid-level feature learning for micro-expression recognition. *Pattern Recognition*, 66:44–52, 2017.
- [45] Jiachi He, Jian-Fang Hu, Xi Lu, and Wei-Shi Zheng. Multi-task mid-level feature learning for micro-expression recognition. *Pattern Recognition*, 66:44–52, 2017.
- [46] Chris House and Rachel Meyer. Preprocessing and descriptor features for facial micro-expression recognition. 2015.
- [47] Xiaohua Huang, Su-Jing Wang, Xin Liu, Guoying Zhao, Xiaoyi Feng, and Matti Pietikainen. Discriminative spatiotemporal local binary pattern with revisited integral projection for spontaneous facial micro-expression recognition. *IEEE Transactions on Affective Computing*, 2017.
- [48] Xiaohua Huang, Su-Jing Wang, Guoying Zhao, and Matti Piteikainen. Facial micro-expression recognition using spatiotemporal local binary pattern with integral projection. In *Proceedings of the IEEE International Conference on Computer Vision Workshops*, page 1–9, 2015.
- [49] Xiaohua Huang, Sujing Wang, Xin Liu, Guoying Zhao, Xiaoyi Feng, and Matti Pietikainen. Spontaneous facial micro-expression recognition using discriminative spatiotemporal local binary pattern with an improved integral projection. *arXiv preprint arXiv:1608.02255*, 2016.

- [50] Xiaohua Huang and Guoying Zhao. Spontaneous facial micro-expression analysis using spatiotemporal local radon-based binary pattern. In *the Frontiers and Advances in Data Science (FADS), 2017 International Conference on*, page 159–164. IEEE, 2017.
- [51] Xiaohua Huang, Guoying Zhao, Xiaopeng Hong, Wenming Zheng, and Matti Pietikäinen. Spontaneous facial micro-expression analysis using spatiotemporal completed local quantized patterns. *Neurocomputing*, 175:564–578, 2016.
- [52] Petr Husák, Jan Čech, and Jiří Matas. Spotting facial micro-expressions" in the wild. In *22nd Computer Vision Winter Workshop*, 2017.
- [53] Rachael E Jack, Oliver GB Garrod, Hui Yu, Roberto Caldara, and Philippe G Schyns. Facial expressions of emotion are not culturally universal. *Proceedings of the National Academy of Sciences*, 109(19):7241–7244, 2012.
- [54] Deepak Kumar Jain, Zhang Zhang, and Kaiqi Huang. Random walk-based feature learning for micro-expression recognition. *Pattern Recognition Letters*, Feb 2018.
- [55] Xitong Jia, Xianye Ben, Hui Yuan, Kidiyo Kpalma, and Weixiao Meng. Macro-to-micro transformation model for micro-expression recognition. *Journal of Computational Science*, 2017.
- [56] Huai-Qian Khor, John See, Raphael Chung Wei Phan, and Weiyao Lin. Enriched long-term recurrent convolutional network for facial micro-expression recognition. In *Automatic Face & Gesture Recognition (FG 2018), 2018 13th IEEE International Conference on*, pages 667–674. IEEE, 2018.
- [57] Dae Hoe Kim, Wissam J. Baddar, and Yong Man Ro. Micro-expression recognition with expression-state constrained spatio-temporal feature representations. In *Proceedings of the 2016 ACM on Multimedia Conference*, page 382–386. ACM, 2016.
- [58] Anh Cat Le Ngo, Alan Johnston, Raphael C.-W. Phan, and John See. Micro-expression motion magnification: Global lagrangian vs. local eulerian approaches. In *Automatic Face & Gesture Recognition (FG 2018), 2018 13th IEEE International Conference on*, page 650–656. IEEE, 2018.
- [59] Anh Cat Le Ngo, Yee-Hui Oh, Raphael C.-W. Phan, and John See. Eulerian emotion magnification for subtle expression recognition. In *Acoustics, Speech and Signal Processing (ICASSP), 2016 IEEE International Conference on*, page 1243–1247. IEEE, 2016.
- [60] Anh Cat Le Ngo, Raphael Chung-Wei Phan, and John See. Spontaneous subtle expression recognition: Imbalanced databases and solutions. In *Asian conference on computer vision*, pages 33–48. Springer, 2014.
- [61] Anh Cat Le Ngo, Raphael Chung-Wei Phan, and John See. Spontaneous subtle expression recognition: Imbalanced databases and solutions. In *Asian conference on computer vision*, page 33–48. Springer, 2014.



- [62] Anh Cat Le Ngo, John See, and C.-W. Raphael Phan. Sparsity in dynamics of spontaneous subtle emotion: Analysis & application. *IEEE Transactions on Affective Computing*, 2017.
- [63] Jingting LI, Catherine Soladié, and Renaud Séguier. Ltp-ml: Micro-expression detection by recognition of local temporal pattern of facial movements. In *Automatic Face & Gesture Recognition (FG 2018), 2018 13th IEEE International Conference on*, pages 634–641. IEEE, 2018.
- [64] Jingting LI, Catherine Soladié, Renaud Séguier, Su-Jing Wang, and Moi Hoon Yap. Spotting micro-expressions on long videos sequences. In *Automatic Face & Gesture Recognition (FG 2019), 2019 14th IEEE International Conference on*. IEEE, 2019.
- [65] Xiaobai Li, Xiaopeng Hong, Antti Moilanen, Xiaohua Huang, Tomas Pfister, Guoying Zhao, and Matti Pietikäinen. Towards reading hidden emotions: A comparative study of spontaneous micro-expression spotting and recognition methods. *IEEE Transactions on Affective Computing*, 2017.
- [66] Xiaobai Li, Tomas Pfister, Xiaohua Huang, Guoying Zhao, and Matti Pietikäinen. *A spontaneous micro-expression database: Inducement, collection and baseline*, page 1–6. IEEE, 2013.
- [67] Xiaobai Li, HONG Xiaopeng, Antti Moilanen, Xiaohua Huang, Tomas Pfister, Guoying Zhao, and Matti Pietikainen. Towards reading hidden emotions: A comparative study of spontaneous micro-expression spotting and recognition methods. *IEEE Transactions on Affective Computing*, 2017.
- [68] Xiaohong Li, Jun Yu, and Shu Zhan. Spontaneous facial micro-expression detection based on deep learning. In *Signal Processing (ICSP), 2016 IEEE 13th International Conference on*, page 1130–1134. IEEE, 2016.
- [69] Yante Li, Xiaohua Huang, and Guoying Zhao. Can micro-expression be recognized based on single apex frame? In *2018 25th IEEE International Conference on Image Processing (ICIP)*, pages 3094–3098. IEEE, 2018.
- [70] Chern Hong Lim and Kam Meng Goh. Fuzzy qualitative approach for micro-expression recognition. In *Proceedings of APSIPA Annual Summit and Conference*, volume 2017, page 12–15, 2017.
- [71] Chenhan Lin, Fei Long, JianMing Huang, and Jun Li. Micro-expression recognition based on spatiotemporal gabor filters. In *2018 Eighth International Conference on Information Science and Technology (ICIST)*, pages 487–491. IEEE, 2018.
- [72] Sze-Teng Liong, Y. S. Gan, Wei-Chuen Yau, Yen-Chang Huang, and Tan Lit Ken. Off-apexnet on micro-expression recognition system. *arXiv preprint arXiv:1805.08699*, 2018.

- [73] Sze-Teng Liong, John See, Raphael C.-W. Phan, Anh Cat Le Ngo, Yee-Hui Oh, and KokSheik Wong. Subtle expression recognition using optical strain weighted features. In *Asian Conference on Computer Vision*, page 644–657. Springer, 2014.
- [74] Sze-Teng Liong, John See, Raphael C.-W. Phan, Yee-Hui Oh, Anh Cat Le Ngo, KokSheik Wong, and Su-Wei Tan. Spontaneous subtle expression detection and recognition based on facial strain. *Signal Processing: Image Communication*, 47:170–182, 2016.
- [75] Sze-Teng Liong, John See, Raphael C.-W. Phan, KokSheik Wong, and Su-Wei Tan. Hybrid facial regions extraction for micro-expression recognition system. *Journal of Signal Processing Systems*, page 1–17, 2017.
- [76] Sze-Teng Liong, John See, Raphael Chung-Wei Phan, and KokSheik Wong. Less is more: Micro-expression recognition from video using apex frame. *arXiv preprint arXiv:1606.01721*, 2016.
- [77] Sze-Teng Liong, John See, KokSheik Wong, Anh Cat Le Ngo, Yee-Hui Oh, and Raphael Phan. Automatic apex frame spotting in micro-expression database. In *Pattern Recognition (ACPR), 2015 3rd IAPR Asian Conference on*, page 665–669. IEEE, 2015.
- [78] Sze-Teng Liong, John See, KokSheik Wong, and Raphael C-W Phan. Less is more: Micro-expression recognition from video using apex frame. *Signal Processing: Image Communication*, 62:82–92, 2018.
- [79] Sze-Teng Liong, John See, KokSheik Wong, and Raphael Chung-Wei Phan. Automatic micro-expression recognition from long video using a single spotted apex. In *Asian Conference on Computer Vision*, page 345–360. Springer, 2016.
- [80] Sze-Teng Liong and KokSheik Wong. Micro-expression recognition using apex frame with phase information. In *Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC), 2017*, page 534–537. IEEE, 2017.
- [81] Yong-Jin Liu, Bing-Jun Li, and Yu-Kun Lai. Sparse mdmo: Learning a discriminative feature for spontaneous micro-expression recognition. *IEEE Transactions on Affective Computing*, 2018.
- [82] Yong-Jin Liu, Jin-Kai Zhang, Wen-Jing Yan, Su-Jing Wang, Guoying Zhao, and Xiaolan Fu. A main directional mean optical flow feature for spontaneous micro-expression recognition. *IEEE Transactions on Affective Computing*, 7(4):299–310, 2016.
- [83] Hua Lu, Kidiyo Kpalma, and Joseph Ronsin. Micro-expression detection using integral projections. 2017.
- [84] Hua Lu, Kidiyo Kpalma, and Joseph Ronsin. Motion descriptors for micro-expression recognition. *Signal Processing: Image Communication*, 2018.

- [85] Zhaoyu Lu, Ziqi Luo, Huicheng Zheng, Jikai Chen, and Weihong Li. A delaunay-based temporal coding model for micro-expression recognition. In *Asian Conference on Computer Vision*, page 698–711. Springer, 2014.
- [86] Iris Lusi, Julio CS Jacques Junior, Jelena Gorbova, Xavier Baró, Sergio Escalera, Hasan Demirel, Juri Allik, Cagri Ozcinar, and Gholamreza Anbarjafari. Joint challenge on dominant and complementary emotion recognition using micro emotion features and head-pose estimation: Databases. In *2017 12th IEEE International Conference on Automatic Face & Gesture Recognition (FG 2017)*, pages 809–813. IEEE, 2017.
- [87] Haoyuan Ma, Gaoyun An, Shengjie Wu, and Feng Yang. A region histogram of oriented optical flow (rhoof) feature for apex frame spotting in micro-expression. In *Intelligent Signal Processing and Communication Systems (ISPACS), 2017 International Symposium on*, pages 281–286. IEEE, 2017.
- [88] Brian W Matthews. Comparison of the predicted and observed secondary structure of t4 phage lysozyme. *Biochimica et Biophysica Acta (BBA)-Protein Structure*, 405(2):442–451, 1975.
- [89] S Mohammad Mavadati, Mohammad H Mahoor, Kevin Bartlett, Philip Trinh, and Jeffrey F Cohn. Disfa: A spontaneous facial action intensity database. *IEEE Transactions on Affective Computing*, 4(2):151–160, 2013.
- [90] Walied Merghani, Adrian Davison, and Moi Hoon Yap. Facial micro-expressions grand challenge 2018: Evaluating spatio-temporal features for classification of objective classes. In *Automatic Face & Gesture Recognition (FG 2018), 2018 13th IEEE International Conference on*, pages 662–666. IEEE, 2018.
- [91] Antti Moilanen, Guoying Zhao, and Matti Pietikäinen. Spotting rapid facial movements from videos using appearance-based feature difference analysis. In *Pattern Recognition (ICPR), 2014 22nd International Conference on*, page 1722–1727. IEEE, 2014.
- [92] Yee-Hui Oh, Anh Cat Le Ngo, Raphael C.-W. Phari, John See, and Huo-Chong Ling. Intrinsic two-dimensional local structures for micro-expression recognition. In *Acoustics, Speech and Signal Processing (ICASSP), 2016 IEEE International Conference on*, page 1851–1855. IEEE, 2016.
- [93] Yee-Hui Oh, Anh Cat Le Ngo, John See, Sze-Teng Liong, Raphael C.-W. Phan, and Huo-Chong Ling. Monogenic riesz wavelet representation for micro-expression recognition. In *Digital Signal Processing (DSP), 2015 IEEE International Conference on*, page 1237–1241. IEEE, 2015.
- [94] Anju P. SureshBabu, Muralidharan K.B, and Arjun K.P. Facial micro-expression recognition using feature extraction. *International Journal of Computer Science and Engineering Communications*, 5(4):1702–1708, Aug 2017.

- [95] Sung Yeong Park, Seung Ho Lee, and Yong Man Ro. Subtle facial expression recognition using adaptive magnification of discriminative facial motion. In *Proceedings of the 23rd ACM international conference on Multimedia*, page 911–914. ACM, 2015.
- [96] Devangini Patel, Guoying Zhao, and Matti Pietikäinen. Spatiotemporal integration of optical flow vectors for micro-expression detection. In *International Conference on Advanced Concepts for Intelligent Vision Systems*, page 369–380. Springer, 2015.
- [97] Min Peng, Chongyang Wang, Tong Chen, Guangyuan Liu, and Xiaolan Fu. Dual temporal scale convolutional neural network for micro-expression recognition. *Frontiers in Psychology*, 8:1745, 2017.
- [98] Min Peng, Zhan Wu, Zhihao Zhang, and Tong Chen. From macro to micro expression recognition: Deep learning on small datasets using transfer learning. In *Automatic Face & Gesture Recognition (FG 2018), 2018 13th IEEE International Conference on*, pages 657–661. IEEE, 2018.
- [99] Min Peng, Zhan Wu, Zhihao Zhang, and Tong Chen. From macro to micro expression recognition: Deep learning on small datasets using transfer learning. In *Automatic Face & Gesture Recognition (FG 2018), 2018 13th IEEE International Conference on*, page 657–661. IEEE, 2018.
- [100] Tomas Pfister, Xiaobai Li, Guoying Zhao, and Matti Pietikäinen. Recognising spontaneous facial micro-expressions. In *Computer Vision (ICCV), 2011 IEEE International Conference on*, pages 1449–1456. IEEE, 2011.
- [101] S Polikovsky. Facial micro-expressions recognition using high speed camera and 3d-gradients descriptor. In *Conference on Imaging for Crime Detection and Prevention, 2009*, volume 6, 2009.
- [102] Senya Polikovsky, Yoshinari Kameda, and Yuichi Ohta. Facial micro-expression detection in hi-speed video based on facial action coding system (facs). *IEICE transactions on information and systems*, 96(1):81–92, 2013.
- [103] David Martin Powers. Evaluation: from precision, recall and f-measure to roc, informedness, markedness and correlation. 2011.
- [104] Fangbing Qu, Su-Jing Wang, Wen-Jing Yan, He Li, Shuhang Wu, and Xiaolan Fu. Cas(me)<sup>2</sup>: a database for spontaneous macro-expression and micro-expression spotting and recognition. *IEEE Transactions on Affective Computing*, 2017.
- [105] Krystian Radlak, Maciej Bozek, and Bogdan Smolka. Silesian deception database: Presentation and analysis. In *Proceedings of the 2015 ACM on Workshop on Multimodal Deception Detection*, pages 29–35. ACM, 2015.
- [106] Matthew Shreve, Sridhar Godavarthy, Dmitry Goldgof, and Sudeep Sarkar. *Macro- and micro-expression spotting in long videos using spatio-temporal strain*, page 51–56. IEEE, 2011.

- [107] Matthew Shreve, Sridhar Godavarthy, Vasant Manohar, Dmitry Goldgof, and Sudeep Sarkar. Towards macro-and micro-expression spotting in video using strain patterns. In *Applications of Computer Vision (WACV), 2009 Workshop on*, page 1–6. IEEE, 2009.
- [108] Patrick A. Stewart, Bridget M. Waller, and James N. Schubert. Presidential speech-making style: Emotional response to micro-expressions of facial affect. *Motivation and Emotion*, 33(2):125, 2009.
- [109] Nicolas Stoiber. *Modeling emotionnal facial expressions and their dynamics for realistic interactive facial animation on virtual characters*. PhD thesis, Université Rennes 1, 2010.
- [110] M. Takalkar and M. Xu. Image based facial micro-expression recognition using deep learning on small datasets. In *The International Conference on Digital Image Computing: Techniques and Applications*. IEEE, 2017.
- [111] Thuong-Khanh Tran, Xiaopeng Hong, and Guoying Zhao. Sliding window based micro-expression spotting: A benchmark. In *International Conference on Advanced Concepts for Intelligent Vision Systems*, pages 542–553. Springer, 2017.
- [112] Alessandro Vinciarelli, Alfred Dielmann, Sarah Favre, and Hugues Salamin. Canal9: A database of political debates for analysis of social interactions. In *Affective Computing and Intelligent Interaction and Workshops, 2009. ACII 2009. 3rd International Conference on*, pages 1–4. IEEE, 2009.
- [113] Su-Jing Wang, Hui-Ling Chen, Wen-Jing Yan, Yu-Hsin Chen, and Xiaolan Fu. Face recognition and micro-expression recognition based on discriminant tensor subspace analysis plus extreme learning machine. *Neural processing letters*, 39(1):25–43, 2014.
- [114] Su-Jing Wang, Bing-Jun Li, Yong-Jin Liu, Wen-Jing Yan, Xinyu Ou, Xiaohua Huang, Feng Xu, and Xiaolan Fu. Micro-expression recognition with small sample size by transferring long-term convolutional neural network. *Neurocomputing*, 2018.
- [115] Su-Jing Wang, Shuhang Wu, and Xiaolan Fu. A main directional maximal difference analysis for spotting micro-expressions. In *Asian Conference on Computer Vision*, page 449–461. Springer, 2016.
- [116] Su-Jing Wang, Shuhang Wu, Xingsheng Qian, Jingxiu Li, and Xiaolan Fu. A main directional maximal difference analysis for spotting facial movements from long-term videos. *Neurocomputing*, 230:382–389, 2017.
- [117] Su-Jing Wang, Wen-Jing Yan, Xiaobai Li, Guoying Zhao, and Xiaolan Fu. Micro-expression recognition using dynamic textures on tensor independent color space. In *Pattern Recognition (ICPR), 2014 22nd International Conference on*, page 4678–4683. IEEE, 2014.

- [118] Su-Jing Wang, Wen-Jing Yan, Xiaobai Li, Guoying Zhao, Chun-Guang Zhou, Xiaolan Fu, Minghao Yang, and Jianhua Tao. Micro-expression recognition using color spaces. *IEEE Transactions on Image Processing*, 24(12):6034–6047, 2015.
- [119] Su-Jing Wang, Wen-Jing Yan, Tingkai Sun, Guoying Zhao, and Xiaolan Fu. Sparse tensor canonical correlation analysis for micro-expression recognition. *Neurocomputing*, 214:218–232, 2016.
- [120] Su-Jing Wang, Wen-Jing Yan, Tingkai Sun, Guoying Zhao, and Xiaolan Fu. Sparse tensor canonical correlation analysis for micro-expression recognition. *Neurocomputing*, 214:218–232, 2016.
- [121] Sujing Wang, Wen-Jing Yan, Guoying Zhao, Xiaolan Fu, and Chunguang Zhou. Micro-expression recognition using robust principal component analysis and local spatiotemporal directional features. In *ECCV Workshops (1)*, page 325–338, 2014.
- [122] Yandan Wang, John See, Yee-Hui Oh, Raphael C.-W. Phan, Yogachandran Rahu-lamathavan, Huo-Chong Ling, Su-Wei Tan, and Xujie Li. Effective recognition of facial micro-expressions with video motion magnification. *Multimedia Tools and Applications*, 76(20):21665–21690, 2017.
- [123] Yandan Wang, John See, Raphael C.-W. Phan, and Yee-Hui Oh. Lbp with six intersection points: Reducing redundant information in lbp-top for micro-expression recognition. In *Asian Conference on Computer Vision*, page 525–537. Springer, 2014.
- [124] Yandan Wang, John See, Raphael C-W Phan, and Yee-Hui Oh. Efficient spatio-temporal local binary patterns for spontaneous facial micro-expression recognition. *PloS one*, 10(5):e0124674, 2015.
- [125] Yandan Wang, John See, Raphael C.-W. Phan, and Yee-Hui Oh. Efficient spatio-temporal local binary patterns for spontaneous facial micro-expression recognition. *PloS one*, 10(5):e0124674, 2015.
- [126] Gemma Warren, Elizabeth Schertler, and Peter Bull. Detecting deception from emotional and unemotional cues. *Journal of Nonverbal Behavior*, 33(1):59–69, 2009.
- [127] Raphaël Weber, Jingting Li, Catherine Soladié, and Renaud Séguier. A survey on databases of facial macro-expression and micro-expression(submitted). *Communications in Computer and Information Science*.
- [128] Raphaël Weber, Catherine Soladié, and Renaud Séguier. A survey on databases for facial expression analysis. In *Proceedings of the 13th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications - Volume 5: VISAPP*, pages 73–84. INSTICC, SciTePress, 2018.
- [129] Qi Wu, Xunbing Shen, and Xiaolan Fu. The machine knows what you are hiding: an automatic micro-expression recognition system. *Affective Computing and Intelligent Interaction*, page 152–162, 2011.

- [130] Zhaoqiang Xia, Xiaoyi Feng, Jinye Peng, Xianlin Peng, and Guoying Zhao. Spontaneous micro-expression spotting via geometric deformation modeling. *Computer Vision and Image Understanding*, 147:87–94, 2016.
- [131] Huang Xiaohua, Su-Jing Wang, Xin Liu, Guoying Zhao, Xiaoyi Feng, and Matti Pietikainen. Discriminative spatiotemporal local binary pattern with revisited integral projection for spontaneous facial micro-expression recognition. *IEEE Transactions on Affective Computing*, 2017.
- [132] Xuehan Xiong and Fernando De la Torre. Supervised descent method and its applications to face alignment. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, page 532–539, 2013.
- [133] Feng Xu, Junping Zhang, and James Z. Wang. Microexpression identification and categorization using a facial dynamics map. *IEEE Transactions on Affective Computing*, 8(2):254–267, 2017.
- [134] Wen-Jing Yan and Yu-Hsin Chen. Measuring dynamic micro-expressions via feature extraction methods. *Journal of Computational Science*, 2017.
- [135] Wen-Jing Yan, Xiaobai Li, Su-Jing Wang, Guoying Zhao, Yong-Jin Liu, Yu-Hsin Chen, and Xiaolan Fu. Casme ii: An improved spontaneous micro-expression database and the baseline evaluation. *PloS one*, 9(1):e86041, 2014.
- [136] Wen-Jing Yan, Su-Jing Wang, Yong-Jin Liu, Qi Wu, and Xiaolan Fu. For micro-expression recognition: Database and suggestions. *Neurocomputing*, 136:82–87, 2014.
- [137] Wen-Jing Yan, Qi Wu, Yong-Jin Liu, Su-Jing Wang, and Xiaolan Fu. *CASME database: a dataset of spontaneous micro-expressions collected from neutralized faces*, page 1–7. IEEE, 2013.
- [138] Shuoqing Yao, Ning He, Huiquan Zhang, and Osamu Yoshie. Micro-expression recognition by feature points tracking. In *Communications (COMM), 2014 10th International Conference on*, page 1–4. IEEE, 2014.
- [139] Moi Hoon Yap, John See, Xiaopeng Hong, and Su-Jing Wang. Facial micro-expressions grand challenge 2018 summary. In *Automatic Face & Gesture Recognition (FG 2018), 2018 13th IEEE International Conference on*, pages 675–678. IEEE, 2018.
- [140] Elham Zarezadeh and Mehdi Rezaeian. Micro expression recognition using the eulerian video magnification method. *BRAIN. Broad Research in Artificial Intelligence and Neuroscience*, 7(3):43–54, 2016.
- [141] Peng Zhang, Xianye Ben, Rui Yan, Chen Wu, and Chang Guo. Micro-expression recognition system. *Optik-International Journal for Light and Electron Optics*, 127(3):1395–1400, 2016.

- [142] Shiyu Zhang, Bailan Feng, Zhineng Chen, and Xiangsheng Huang. Micro-expression recognition by aggregating local spatio-temporal patterns. In *International Conference on Multimedia Modeling*, page 638–648. Springer, 2017.
- [143] Xing Zhang, Lijun Yin, Jeffrey F Cohn, Shaun Canavan, Michael Reale, Andy Horowitz, Peng Liu, and Jeffrey M Girard. Bp4d-spontaneous: a high-resolution spontaneous 3d dynamic facial expression database. *Image and Vision Computing*, 32(10):692–706, 2014.
- [144] Zhihao Zhang, Tong Chen, Hongying Meng, Guangyuan Liu, and Xiaolan Fu. Smeconvnet: A convolutional neural network for spotting spontaneous facial micro-expression from long videos. *IEEE Access*, 6:71143–71151, 2018.
- [145] Yue Zhao and Jiancheng Xu. Necessary morphological patches extraction for automatic micro-expression recognition. *Applied Sciences*, 8(10):1811, 2018.
- [146] Hao Zheng. Micro-expression recognition based on 2d gabor filter and sparse representation. In *Journal of Physics: Conference Series*, volume 787, page 012013. IOP Publishing, 2017.
- [147] Hao Zheng, Xin Geng, and Zhongxue Yang. A relaxed k-svd algorithm for spontaneous micro-expression recognition. In *Pacific Rim International Conference on Artificial Intelligence*, page 692–699. Springer, 2016.
- [148] Xuena Zhu, Xianye Ben, Shigang Liu, Rui Yan, and Weixiao Meng. Coupled source domain targetized with updating tag vectors for micro-expression recognition. *Multimedia Tools and Applications*, page 1–20, 2017.
- [149] Yuan Zong, Xiaohua Huang, Wenming Zheng, Zhen Cui, and Guoying Zhao. Learning a target sample re-generator for cross-database micro-expression recognition. In *Proceedings of the 2017 ACM on Multimedia Conference*, pages 872–880. ACM, 2017.
- [150] Yuan Zong, Xiaohua Huang, Wenming Zheng, Zhen Cui, and Guoying Zhao. Learning from hierarchical spatiotemporal descriptors for micro-expression recognition. *IEEE Transactions on Multimedia*, 2018.
- [151] Yuan Zong, Wenming Zheng, Xiaohua Huang, Jingang Shi, Zhen Cui, and Guoying Zhao. Domain regeneration for cross-database micro-expression recognition. *IEEE Transactions on Image Processing*, 27(5):2484–2498, 2018.





# List of Figures

0-1	Tendance de la recherche MESR. Le nombre d'articles sur MESR augmente d'année en année, principalement dans le domaine de la reconnaissance de l'ME (colonne du bas). La recherche sur la détection de ME n'a pas encore suffisamment attiré l'attention (colonne en haut). . . . .	4
0-2	Exemple de motif temporel local (LTP) au cours d'une ME située dans la région du sourcil droit sur une période de 300 ms (durée moyenne de ME). Ce LTP représente l'évolution de la texture (niveau de gris) d'une ROI pendant la ME. Il forme un motif en S (S-pattern). La courbe atteint son sommet au bout de 150 ms environ, puis reste stable ou diminue légèrement. Ce motif est spécifique aux mouvements ME et est appelé motif en S (S-pattern) en raison de la forme de la courbe.(Vidéo: Sub01_EP01_5 of CASME I) . . . . .	6

- 0-3 Vue d'ensemble de notre méthode. La méthode proposée comporte trois étapes de traitement: pré-traitement, extraction de caractéristiques et détection de micro-expressions. Nous mélangeons des processus globaux (tout le visage) et locaux (les ROIs). La sous-étape d'extraction des caractéristiques et la première sous-étape de la détection de la micro-expression sont effectuées sur des régions d'intérêt (ROIs) pertinentes, tandis que les autres étapes sont réalisées sur tout le visage (global). Les LTPs, y compris les S-patterns, sont ensuite utilisés comme échantillons d'apprentissage pour construire le modèle d'apprentissage automatique (SVM) en vue de la classification. En particulier, une fusion spatiale et temporelle finale est réalisée pour éliminer les faux positifs tels que les clignements des yeux. L'une des spécificités du processus réside dans l'utilisation de motifs temporels locaux (LTP), pertinents pour la détection de micro-expressions: les micro-expressions sont des mouvements brefs et locaux. . . . . 10
- 0-4 ACP sur l'axe des temps par ROI. Une séquence vidéo locale  $ROI_j$  avec  $N$  images (durée de la vidéo  $\leq 3s$ ) est traitée par l'ACP sur l'axe des temps. Les premières composantes de l'ACP conservent le mouvement principal de la texture de niveau de gris sur cette ROI pendant cette durée ( $N$  images). L'échantillon vidéo provient de CASME I (©Xiaolan Fu) . . . . . 11
- 0-5 Sélection de LTP pour l'étape d'entraînement. Tous les LTPs sont classés en 2 classes: S-patterns et non-S-patterns. Les LTPs passent par 3 étapes pour l'annotation: l'annotation par image, la sélection de ROI à partir de l'AU et la sélection du motif de LTP. Les S-patterns annotés et les non-S-patterns sont ensuite transmis à l'étape d'apprentissage du classifieur SVM. 13

0-6 Filtrage de LTP et synthèse du S-pattern par le modèle d'Hammerstein (HM) pendant la phase d'entraînement. Dans le bloc de droite, le S-pattern original ( $S\text{-pattern}_O$ ), après l'annotation et la sélection de l'AU de la figure 0-5) passe par l'identification du système du modèle d'Hammerstein. L'ensemble de paramètres  $(\alpha, \beta, E_H)$  correspondant à ce S-pattern est ensuite estimé. Les S-patterns sont sélectionnés par le processus de filtrage de LTP en fonction de l'erreur d'estimation  $E_H$ . Les motifs sélectionnés ( $S\text{-pattern}_{OF}$ ) sont utilisés pour générer  $n$  S-patterns synthétisés ( $S\text{-patterns}_{ST}$ ). Pour la comparaison, le bloc de gauche montre notre méthode sans modèle d'Hammerstein, c'est-à-dire le résultat après la sélection du motif de LTP:  $S\text{-pattern}_{OS}$ .

Les abréviations ci-dessous sont fréquemment utilisées.  
 $S\text{-pattern}_O$ : S-pattern original après annotation de l'étiquette et sélection de l'AU de la Figure 0-5.  
 $(\alpha, \beta)$ : paramètres dans le module linéaire du modèle d'Hammerstein.  
 $E_H$ : Erreur d'estimation du modèle d'Hammerstein  
 $S\text{-pattern}_{OF}$ : S-pattern original ( $S\text{-pattern}_O$ ) conservé après le filtrage de LTP.  
 $S\text{-pattern}_{ST}$ : S-pattern synthétisé par le modèle d'Hammerstein.  
 $S\text{-pattern}_{OS}$ : S-pattern original ( $S\text{-pattern}_O$ ) conservé après la sélection du motif LTP de la Figure 0-5. . . . . 15

1-1 MESR research trend. The number of articles on MESR is increasing by year, mainly in the area of ME recognition (bottom column). ME spotting research has not yet attracted sufficient attention (column at the top). . . . . 28

1-2	Example of Local temporal pattern (LTP) during a ME located in the right eyebrow region over a period of 300ms (average duration of ME). This LTP represents the evolution of the grey level texture of one ROI during the ME. The curve reaches the top in around 150ms and then stays stable or slightly declines. This pattern is specific of ME movements, and is referred to S-pattern due to the curve shape. (Video: Sub01_EP01_5 of CASME I) . . . .	30
2-1	Samples in spontaneous micro-expression database . . . . .	42
2-2	Three Modalities of SMIC database. Sample at the left side is NIR image, in the middle is the VIS image and at the right side is the HS image. . . . .	46
2-3	Images samples from MEVIEW database . . . . .	50
2-4	Acquisition setup for elicitation and recording of micro-expressions [135] .	51
2-5	Histogram of action units (AUs) annotations for ME databases. The AU amount on the region of eyebrows (e.g. AU 1,2,4) and mouth (e.g. AU 12, 14) indicate that these two regions have the most frequent ME movement. .	52
2-6	Micro-expression spot-and-recognize scheme. 2-6a: the non-micro-expressions are identified by recognition method. 2-6b:the micro-expression samples are firstly spotted in long videos, then they are classified into different emotion classes by recognition methods. . . . .	74
3-1	Overview of our method. The proposed method contains three steps of processing: pre-processing, feature extraction and micro-expression spotting. We mix both global and local processes. Both sub-step of feature extraction and the first sub-step of micro-expression spotting are performed on relevant regions of interest (ROIs). LTPs including S-patterns are then used as the training samples to build the machine learning model (SVM) for classification. Especially, a final spatial and temporal fusion is performed to eliminate the false positives such as eye blinks. The specificity of the process is the use of local temporal patterns (LTP), which are relevant for micro-expression spotting: micro-expressions are brief and local movements.	80

3-2	Facial landmarks and ROIs distribution. 49 landmarks are detected and ROIs are generated depending on the position of 12 chosen landmarks. 12 ROIs, relevant for micro-expression, are selected from the region of eyebrows, nose and mouth contour.(©Xiaolan Fu) . . . . .	82
3-3	PCA on time axis per ROI. A local video sequence of ROI <sub>j</sub> with <i>N</i> frames (video duration $\leq 3$ s) is processed by the PCA on the time axis. The first components of PCA conserve the principal movement of grey level texture on this ROI in this duration ( <i>N</i> frames). The video sample comes from CASME I (©Xiaolan Fu) . . . . .	83
3-4	PCA energy analysis. The movement is contained in the first 2 components with more than 80% energy. (Sub01_EP03_5_ROI10 of CASMEI) . . . . .	84
3-5	Interface of the result distribution after PCA process. (3-5a) The point distribution corresponds to all frames in one ROI sequence in PCA projection space. The chosen ROI is the inner side of left eye brow. The blue, red and green dots represent the frames before onset, from onset to offset and after offset respectively, and the yellow dots mean the apex frames. (3-5b, 3-5c and 3-5d) Displacement comparison by ROI images. The first image illustrate the first frame in the ROI sequence. The three images on the right represent the current frame (CF), CF-1 and CF+1. The red arrow shows the displacement of the eyebrow. In 3-5b, there is barely no movements at the beginning of the sequence. In 3-5c, the eyebrow goes down due to the micro-expression. In 3-5d, the eyebrow is raised up compared with apex frame because the ME fades out. Besides, the position of eyebrow is even higher than the first frame due to other facial movements. Comparing the arrow length in the different frames, a conclusion can be obtained, i.e. geometric features in the 3-5a depending on the distribution of frames can represent the temporal displacement in ROI. . . . .	86

3-6	Extraction of local temporal pattern. In the sequence $ROI_j$ , the $n$ th frame can be represented as point $P_n^j$ in the PCA projection space. The frame distribution on the PCA projection space of this video sequence is illustrated in 3-6a.(Continued on the next page) . . . . .	88
3-6	The larger the distance between two points is, the more evident the displacement on ROI region between these two frame is. 3-6b shows the point distribution from $P_n^j$ to $P_{n+K}^j$ (frames from $n$ to $n + K$ , 300ms). Normalized distances between this $n$ th frame and other $K$ following frames are calculated as shown in 3-6c. The ensemble of distances forms a curve, which is called the local temporal pattern (LTP). And here is the S-pattern for the ME frames. Meantime, 3-6d shows the point distribution from $P_m^j$ to $P_{m+K}^j$ (Non-ME frames from $m$ to $m + K$ ). The corresponding LTP pattern is illustrated in 3-6e. . . . .	89
3-7	Histogram for normalization coefficient (CN) value for all ROI sequences in CASME I. X axis means the CN value, Y axis is the ROI sequence amount for each bin. The average CN value is around 4. Yet, there is a few ROI sequences which have CN values larger than 10, which represents there is almost none evident movement in this video. In the feature construction step, the CN value for this kind of ROI sequences is set to 10. . . .	91
3-8	Local temporal patterns (LTPs) of two different videos.(Continued on the next page) . . . . .	93

3-8	Local temporal patterns (LTPs) of two different videos. 3-8a and 3-8b are the LTPs during the micro-expression movement at ROI 32 and ROI 38 (the right and left corners of the mouth) in the video Sub01_EP01_5. The emotion of this video is labeled as joy, which is often expressed by mouth corner displacement. 3-8e and 3-8f are the LTPs during the micro-expression movement at ROI 5 and ROI 6 (the inside corners of the right and left eyebrows) in the video Sub04_EP12. The emotion of this video is labeled as stress, which is often expressed by the movement of the eyebrows. The pattern of curves in these four images are similar even through the ROIs and subjects are different, we call it S-pattern. 3-8c and 3-8g show the LTP of other ROIs at the same time as 3-8a/ 3-8b and 3-8e/ 3-8f respectively. The pattern is different from S-pattern because the micro-expression does not occur on these regions. 3-8d and 3-8h illustrate the LTPs in the same ROI as 3-8a and 3-8f respectively, but at a different moment in the video. These patterns differ from S-pattern since the micro-expression does not occur at this moment. (video samples in CASME I) . . . . .	94
3-9	LTP selection for training stage. All the LTPs are classified into 2 classes: S-patterns and non-S-patterns. LTPs pass through 3 steps for the annotation: label annotation per frame, AU selection per ROI and LTP pattern selection. The annotated S-patterns and non-S-patterns are then fed to the training stage of the SVM classifier. . . . .	95
3-10	Label annotation per frame. The rectangle represents an entire video, and the interval from onset to offset is the ME sequence. The S-pattern is expressed at the beginning of the onset. Hence, frames with S-pattern (in the range [onset- $K/3$ , onset]) are labeled with label 1 (S-pattern) and the other frames are labeled with 0 (non S-pattern). . . . .	96
3-11	LTP pattern selection of training stage for local classification. LTPs labeled as S-pattern pass through this process to conserve reliable S-patterns. The selection criteria include distance value $d$ in LTP, normalization coefficient ( $CN$ ) and curve slope( $p_{LTP}$ ). . . . .	97



- 3-12 Spatial and temporal fusion. The predicted labels from local classification for all ROIs are represented as  $LC_{ROI_{j=1,\dots,J}}^{n=1,\dots,N}$ , where  $N$  is the number of frames in the whole video,  $J$  is the chosen ROIs amount. Passing through the local qualification (LQ) per ROI sequence, the spotting intervals which are too short or too long are deleted. Then for each frame  $n$ , the local spotting results  $LQ_{ROI_{j=1,\dots,J}}^n$  are integrated into a single  $SF_n$ , which represents the spotting result for the entire frame. A merge process is applied on  $SF_{n=1,\dots,N}$  to form a consecutive ME movement. Thus, we get the final result  $STF_{n=1,\dots,N}$  for one video sequence. . . . . 98
- 3-13 Flow chart of temporal selection process in local qualification. The  $j_{th}$  ROI sequence passes through the distance and normalization coefficient threshold selection, then predicted label of this sequence  $LC_{ROI_j}^{n=1,\dots,N}$  enter the process as input.  $LQ_{ROI_j}^{n=1,\dots,N}$  is the output of the process, i.e. the result after local qualification (LQ). . . . . 99
- 3-14 Flow chart of spatial fusion.  $I_{nose}$  means the all the ROI index on nose region, and  $I_{eyebrow}$  means all the ROI index on eyebrows. For  $n_{th}$  frame in video sample, predicted label of all  $J$  ROIs  $LQ_{ROI_{j=1,\dots,J}}^n$  enter the spatial fusion process. The output  $SF_n$  is a predicted label for this frame, and it represents that whether there is micro-expression on this entire frame or not. 100
- 3-15 Flow chart of merge process. . . . . 101
- 3-16 The long video is divided into several short sequences ( $I_1, \dots, I_m, \dots, I_M$ ) by a sliding window (1s). . . . . 103
- 3-17 Facial landmarks tracking and ROI selection. On the right: an example from SAMM; on the left: an example from CAS(ME)<sup>2</sup> . . . . . 104

4-1 Overview of our method combining Hammerstein model. Our proposed method has been presented in Figure 3-1. The grey block replaces the LTP selection by using Hammerstein Model. More reliable S-patterns (LTP patterns specific to ME movements) are produced by this model. LTPs including S-patterns (both real and synthesized) are then used as the training samples to build the machine learning model (SVM) for classification. . . . 108

4-2 LTP filtering and S-pattern synthesizing by Hammerstein model (HM) during the training stage. In the right block, the original S-pattern (S-pattern<sub>O</sub>, after label annotation and AU selection in Figure 3-9) passes through the system identification of Hammerstein model. The parameter set ( $\alpha, \beta, E_H$ ) corresponding to this S-pattern is estimated. S-patterns are selected by LTP filtering process according to the estimation error  $E_H$ . The selected patterns (S-pattern<sub>OF</sub>) are used to generate  $n$  synthesized S-patterns (S-patterns<sub>ST</sub>). For comparison, the left block shows our method without Hammerstein model, i.e. the result after LTP pattern selection: S-pattern<sub>OS</sub>.

The below abbreviation are frequently used:  
 S-pattern<sub>O</sub>: original S-pattern after label annotation and AU selection of Figure 3-9.  
 ( $\alpha, \beta$ ): parameters in the linear module of Hammerstein model.  
 $E_H$ : Estimation error of Hammerstein model.  
 S-pattern<sub>OF</sub>: Conserved S-pattern<sub>O</sub> after LTP filtering.  
 S-pattern<sub>ST</sub>: Synthesized S-pattern by Hammerstein model.  
 S-pattern<sub>OS</sub>: Conserved S-pattern<sub>O</sub> after LTP pattern selection of Figure 3-9. . . . . 109

4-3 Hammerstein model structure. Hammerstein model represents well the muscle movement. S-patterns are caused by the facial muscle movement and the variation curve is similar to the local muscle movement. Thus, S-patterns can be well synthesized by Hammerstein model. The model is a concatenation of two modules: a static non-linearity module (that manipulates the magnitude) and a second-order dynamics linear module (that simulates the movement pattern). . . . . 110

4-4 The basic system identification process for Hammerstein model. Depending on the data which is constructed by constant command and chosen S-pattern<sub>O</sub>, the corresponding Hammerstein model can be estimated by system identification. In other words, the parameters of the non-linearity module ( $p$ ), of the linear module ( $\alpha, \beta$ ) and the system estimation error ( $E_H$ ) are determined. . . . . 112

4-5 One Example of real S-pattern (S-pattern<sub>O</sub>) and the corresponding synthesized S-pattern (S-pattern<sub>ST</sub>) by estimated by Hammerstein model. . . . . 113

4-6 The distribution of  $(\alpha, \beta)$  is related to the curve shape of S-pattern. Each S-pattern ( $S\text{-pattern}_{OS}$ ) has been associated with its own identified Hammerstein model  $(\alpha, \beta, E_H)$ . The upper-left figure shows the distribution of  $(\alpha, \beta)$  ( $x: \alpha, y: \beta$ ) and the associated error:  $E_H$  (heat map).  $(\alpha, \beta)$  densely distributes at the top-left corner with a small error. In the six below images, the blue curve means the original S-patterns ( $S\text{-pattern}_{OS}$ ). Then, based on the estimated Hammerstein model with original  $(\alpha, \beta)$ , the synthesized S-patterns ( $S\text{-pattern}_{ST_0}$ ) are generated (red curve). The curve shape of S-patterns in these six figures vary along with the change of  $(\alpha, \beta)$ . The first three curve images correspond to the densely distributed region of  $(\alpha, \beta)$ . The corresponding  $(\alpha, \beta)$  for the last three curve images are far from the upper-left region. They have different curve shapes compared with the first three. The distribution of  $(\alpha, \beta)$  is associated with the dynamic property of ME (shape of S-pattern). Hence, we would be able to both filter wrongly-labeled  $S\text{-patterns}_O$  using  $E_H$  values and also synthesize virtual S-patterns based on the value range of  $(\alpha, \beta)$ . . . . . 116

4-7 Parameter configuration for LTP filtering and S-pattern synthesizing. For one database, each selected S-pattern is treated separately to estimate its specific Hammerstein model. Based on these obtained data, the mean value  $(\bar{\alpha}, \bar{\beta}, \bar{E}_H)$  and the standard deviation  $(\sigma_\alpha, \sigma_\beta)$  can be calculated. . . . . 117

4-8 Flow chart of LTP filtering process. The original  $S\text{-pattern}_O$  dataset may contain some wrongly-labeled samples. Each original S-pattern ( $S\text{-pattern}_O$ ) passes through the system identification to obtain its estimation error  $E_H$ . By comparing with the threshold  $T_E$ , the  $S\text{-pattern}_O$  is decided to be kept as S-pattern after LTP filtering ( $S\text{-pattern}_{OF}$ ) or be removed from training dataset. . . . . 118

4-9	Examples of eliminated real LTPs (labeled as S-patterns) after S-pattern filtering. The threshold $T_E$ is set to 0.0250. The curve shape in 4-9a represents a movement which begins to fade out. 4-9b and 4-9c show facial movements which are about to begin. 4-9d is the movement at the end of video sequence. These patterns are removed from training set by LTP filtering. . . . .	120
4-10	Flow chart of S-pattern synthesizing. The number of generation loops is defined by $n$ . Once the $(\alpha_i, \beta_i)$ is determined, along with the S-pattern <sub>OF</sub> , the specific Hammerstein model is constructed for synthesizing. S-pattern <sub>OF</sub> is needed in this step because it helps to identify the parameters $p$ in the non-linear module. Then the S-pattern <sub>ST<sub>i</sub></sub> is synthesized by the Hammerstein model <sub><math>i</math></sub> whose input is the constant command $u(t)$ . . . . .	122
4-11	Example of 1 original S-pattern (S-pattern <sub>O</sub> ) and 10 S-patterns generated by Hammerstein model (S-patterns <sub>ST</sub> ). Depending on the S-pattern <sub>O</sub> , we can generate $n$ times similar S-patterns <sub>ST</sub> for data augmentation. . . . .	122
5-1	Organization of Chapter5 based on our four contributions. . . . .	125
5-2	Baseline method -LBP $\chi^2$ -distance difference. In the first step, the face image is divided into several blocks. Then LBP is computed per pixel. The feature for the $j$ th block on current frame (CF) is the LBP histogram per ROI after a normalisation. The second step is $\chi^2$ -distance computation. $i$ is the $i$ th bin in the histogram. AFF (Average feature frame) means the feature vector representing the average of tail frame and head frame, where tail frame is $K/2$ th frame before the CF, head frame is $K/2$ th frame after the CF. The third step is to obtain the final feature difference value $C_{CF}$ for current frame. $F$ is obtained by the first $M$ blocks with the biggest feature difference values. The fourth step spots micro-expression by setting a threshold, where $p$ is an empirical data. . . . .	130
5-3	LBP-TOP extraction per ROI. LBP feature is extracted from three orthogonal planes: xy, xt and yt. . . . .	136

5-4 Influence of ROI size on Spotting performance. The F1-score increases along with the augmentation of ROI size before the region length reaches 20. It is because that the information in regions which are too small is not enough to represent the micro-expression local movement. Yet, the spotting performance is then affected when the ROI size gets larger than 20. More irrelevant information is included in the region for analysis. It raise the false positive in the final analysis. . . . . 140

5-5 Auto-encoder system for dimension reduction of ROI sequence on time axis ( $N$ ). The auto-encoder system can encode the ROI sequence ( $a^2 * N$ ) into a 2D dimension point distribution, and then reconstruct the input by decoder process.  $a$  is the ROI width,  $a^2$  is the pixel amount in one ROI image, and  $N$  means the frame amount in this ROI sequence. The frame index represents the temporal relation in this video. . . . . 142

5-6 Dimension reduction schema by Auto-encoder. The process contains two stages: AE training stage and dimension reduction. For each video sample, all  $J$  ROI sequences are concatenated into one matrix in size of  $(a^2 * N) * J$ . To conserve the principal variation, the matrix is normalized for AE training. Then, the obtained encoder can extract the maximal movement on time axis of the chosen ROI sequence. Like PCA process, the dimension of the input data is reduced to 2 dimensions. Thus,  $N$  2D points which represent the ROI sequence can be obtained in AE projection space. . . . . 142

5-7 Comparison between LTP examples obtained by PCA, AE and GPLVM. For the same micro-expression sequence, these two dimension reduction methods can obtain similar LTP patterns. (CASMEI-A, sub1\_EP07\_10, ROI5) . . . . . 143

5-8 Example of global result obtained by each step in spatial and temporal fusion. The X axis is the frames index, the Y axis is the predicted label, and the red curve represents the ground truth for this video. As introduced in Figure 3-12, local classification and qualification are applied on ROIs. The first and second layers in the figure show the result obtained directly by the local classification (LC) and result after local qualification (LQ). The different colored lines represent the spotting results per ROI. The third and fourth layers give the global results after separately applying spatial fusion (SF) and merge process (MP) over the LC global result (blue curve). And the fifth layer is the final result after the three spatial and temporal fusion steps (STF). (CASME I\_Sub08\_EP12\_2\_1) . . . . . 146

5-9 GAN structure for S-pattern synthesizing. Ransom noise is used as the input, the generator is trained to generate S-patterns, and the discriminator will compare the synthesized pattern with S-patterns from databases. Generator and discriminator are two shallow convolution networks. . . . . 151

5-10 Synthesized S-pattern samples for CASME I-section A by GAN. Only 5-10a represents well the S-pattern. The other three LTPs can not be treated as reliable S-patterns. 5-10b and 5-10c show the movement with onset too long or too short. 5-10d is a on-going movement which has longer duration than micro-expression. . . . . 152

5-11	Result Evaluation according to the generation times $n$ for the combination of LTP filtering and S-pattern synthesizing on CASME II. $T_E$ is set to 0.25 for LTP filtering. 5-11a shows the ratio between S-pattern and the total quantity of LTPs for training. And 5-11b illustrates the F1-score for ME spotting. The parameter $n$ for S-pattern synthesizing stops at 5 for CASME II, because the amount of S-patterns in training dataset is large enough and the F1-score have already begun to decline. By comparing these two figures, when the proportion of S-patterns is around 0.3, the spotting method performs best. Otherwise, the data augmentation process also synthesizes more wrongly-labeled S-patterns. It would import extra noise into the training process and then influence the spotting performance. . . . .	155
5-12	Result Evaluation according to the error threshold value for LTP filtering process. 5-12a shows the data augmentation of the S-pattern in the training dataset. More S-patterns are conserved while the threshold is larger. The increasing trend stops when the threshold can not filter any patterns. Unlike the curve for S-pattern amount, in 5-12b the curve of F1-score $_{fr}$ increases at the beginning when there are more samples for training, then it starts to decline as the filtering process conserves too many wrongly-labeled S-patterns. . . . .	156
5-13	Improvement of F1-score $_{fr}$ by increasing the training set volume with synthesized S-patterns from Hammerstein model. The result is evaluated depending on the S-pattern amount. x axis means the generation multiple $n$ of S-pattern $_O$ . Y axis is the F1-score. As the quantity of S-pattern increases, the F1-score $_{fr}$ value becomes higher than that of S-pattern $_OF$ . . . . .	157
5-14	Histogram of $\alpha$ and $\beta$ of the linear model for S-pattern . . . . .	158
C-1	Distance calculation for one ROI sequence $ROI_j^m$ in video clip $I_m$ . . . . .	180





# List of Tables

2.1	Database amount for macro- and micro-expression. . . . .	36
2.2	Reference of published micro-expression databases. . . . .	36
2.3	Characteristics summary for micro-expression databases - part 1. Databases are sorted by alphabetical order. The following formatting distinguishes databases: normal for posed databases, bold for spontaneous database, italic for in-the-wild databases, †means that the database is not available online. MaE: macro-expression, PSR: participants' self-report. . . . .	38
2.4	Characteristics summary for micro-expression databases - part 2. Databases are sorted by alphabetical order. The following formatting distinguishes databases: normal for posed databases, bold for spontaneous database, italic for in-the-wild databases, †means that the database is not available online. Neutralization paradigm: NP, PD:Polikovsky's Database, GD: Grobova's database, PSR: participants' self-report. . . . .	39
2.5	Characteristics summary for micro-expression databases - part 3. The following formatting distinguishes databases: normal for posed databases, bold for spontaneous database, italic for in-the-wild databases. PSR: participants' self-report. HS: high speed camera, VIS: a normal visual camera, NIR: a near-infrared, . . . . .	40
2.6	Categories and characteristics of ME databases. The characteristics are coded to simplify further representation. . . . .	43

2.7	Classification of the databases according to the characteristic P.1, P.2, H.1, A.1 and A.2 (# of subjects, # of samples, modalities, action units and emotional labels). Databases are sorted by alphabetical order. The following formatting distinguishes databases: normal for posed databases, bold for spontaneous database, italic for in-the-wild databases, * means the database is not available online. 2D V: 2D video. SMIC and SMIC-E both have three sub-classes: NIR, VIS and HS. Sub-class HS of SMIC / SMIC-E is separated from the other two because of the different number of ME video samples. . . . .	45
2.8	Number of frames for a micro-expression with average duration (300ms) depending on different FPS. . . . .	47
2.9	Emotion classes and sample numbers for micro-expression databases. . . .	53
2.10	The number and frequency of ME recognition articles, according to the metrics used. CM means the confusion matrix, time includes the training time, recognition time and computation/run time. . . . .	58
2.11	Summary of number of published articles, with their corresponding metrics and databases. CAS I: CASME I; CAS II: CASME II; SMIC includes SMIC and SMIC-E. ACC: accuracy, CM: confusion matrix. . . . .	58
2.12	Summary of emotion classes for micro-expression recognition. The two most commonly used emotion classes are highlighted in bold. P: positive, N: negative, H: Happiness, D: Disgust, SU: Surprise, R: Repression, T: Tense, F: Fear, C: Contempt, SA: Sadness. . . . .	61
2.13	Published spontaneous micro-expression spotting per frame methods and their corresponding general metrics. The method highlighted in bold is the most commonly used method for comparison. DF.1: apex/onset spotting methods, DF.2 : feature difference methods, DF.3: machine learning methods. ACC: accuracy; ROC: receiver operating characteristic curve; TPR: true positive rate; AE: average error, and this column also includes the articles which used MAE (mean absolute error). . . . .	64

2.14	Database numbers and frequency (%) for ME spotting. The number of two most frequently used databases are highlighted in bold. CAS II : CASME II, CAS I ; CASME I. . . . .	66
2.15	The number and frequency of ME spotting articles, according to the metrics used. ACC: accuracy, AE: average error, AE also includes MAE. . . . .	67
3.1	Chosen ROIs and related AU index. The ROI index is annotated depending on the detected facial landmarks. 12 ROIs mean that the local regions which have most evident motion for micro-expressions are selected. . . . .	82
3.2	Distance sets per frame for one ROI video sequence . . . . .	90
3.3	ROI selection for training stage of machine learning. Since micro-expression is a local movement, ROIs which have annotated AU are chosen to represent the micro-expression movement. . . . .	96
4.1	Estimation performance of different non-linearity modules in Hammerstein model. Fit rate is the fit rate of generated pattern to original pattern; Loss-Fcn means the result of loss function. All the estimators are performed with default parameters. . . . .	115
5.1	Main parameters and experiment parameters configuration for CASME I (Section A and B), CASME II, CAS(ME) <sup>2</sup> and SAMM. . . . .	131
5.2	Result evaluation and comparison for LTP-ML and LBP- $\chi^2$ . LTP-ML (without merge) can spot more TP frames with an acceptable FPR. In addition, LTP-ML outperforms state-of-art LBP- $\chi^2$ method in terms of F1-score in both cases: with or without merge process. Merge process helps to reduce the true negatives. (F1-score <sub>fr</sub> : F1-score per frame) . . . . .	135
5.3	LTP outperforms LBP-TOP on both spotting accuracy and computation time. <i>Time<sub>extract</sub></i> means the average time for feature extraction per one video sample with 200 frames, and <i>Time<sub>svm</sub></i> means the average time for SVM training and classifying per video. (CASME I-A) . . . . .	136

5.4	Cross database ME spotting performance shows the generality of LTP among different databases. The cross-database result is evaluated by $F1\text{-score}_{fr}$ . The experiments in the same database is performed by leave-one subject-out cross validation. . . . .	137
5.5	Spotting results per emotion on CASME I . . . . .	138
5.6	S-pattern differs from other LTP (non-S-patterns). The LTP is statistically analyzed by the average value and the standard deviation of the following two characteristics: the maximal value of S-pattern ( $D_{max}$ ) and the curve slope of S-pattern in first 150ms ( $Slope_{150ms}$ ). . . . .	139
5.7	Unique S-pattern for different emotions. The S-pattern is statistically analyzed by the average value and the standard deviation of the following two characteristics: the maximal distance value in S-pattern ( $Dist_{max}$ ) and the curve slope of S-pattern in first 150ms ( $Slope_{150ms}$ ). . . . .	139
5.8	Two situations of ROI index. The ROI index is annotated depending on detected facial landmarks. . . . .	141
5.9	Spotting performance evaluation (F1-score) based on different amount of chosen ROIs. . . . .	141
5.10	Spotting performance evaluation after PCA and AE process. The F1-score of AE is slightly higher than that of PCA. And the GPLVM does not improve much the spotting performance. Yet, the time for feature extraction from the entire database by AE and GPLVM is much longer than PCA. PCA is more suitable for spotting micro-expressions in real time. . . . .	144
5.11	Analysis of each step for spatial and temporal fusion (STF) on CASME I-A. LC: local classification result; LQ: local qualification; SF: spatial fusion and MP: merge process. The decreasing of FP amount shows that LQ and SF process helps to reduce the irrelevant facial movements. More TP frames are spotted due to the merge process. A combination of these three steps keeps the spotting performance and largely reduces the false positives. . . . .	147
5.12	Impact of $T_{dist}$ on the fusion process. . . . .	148

5.13	Impact of $T_{CN}$ on the fusion process. $\overline{CN}$ means the average value of normalization coefficient for each ROI sequence. . . . .	148
5.14	Result evaluation of LTP-SpFS and comparison with state-of-art method. $(LBP-\chi^2)^+$ method represents the $LBP-\chi^2$ with a supplementary merge process. $F1\text{-score}_{fr}$ means F1-score of an evaluation per frame; $F1\text{-score}_I$ means the metric proposed by MEGC (F1-score of an evaluation per interval). Our proposed LTP-SpFS method improves the spotting performance of LTP-ML and outperforms the SOA $LBP-\chi^2$ method in terms of F1-score (both metrics). . . . .	150
5.15	S-patterns synthesized by Hammerstein model outperform that generated by GAN for micro-expression spotting. . . . .	153
5.16	ME spotting result in terms of $F1\text{-score}_{fr}$ with data augmentation by Hammerstein model. LTP-ML represents our method without Hammerstein model (HM); LTP-SpF is our method with HM but only LTP filtering; LTP-SpS is our method with HM but only S-pattern synthesizing; LTP-SpFS represents the whole process with Hammerstein model (LTP filtering + S-pattern synthesizing). The spotting results for SpF and SpS are better than LTP-ML in CASME I because the size of S-pattern dataset for training stage is increased. In addition, the combination of these two steps improves the spotting performance due to a larger data volume of S-pattern. . . . .	154
5.17	Spotting performance comparison between S-pattern synthesizing by normal distribution and Poisson distribution. . . . .	158
5.18	Parameter configuration for SAMM and CAS(ME) <sup>2</sup> depending on FPS and faical resolution. $L_{window}$ is the length of sliding window $W_{video}$ , $L_{overlap}$ is the overlap size between sliding windows, $L_{interval}$ is the interval length of $W_{ROI}$ . The facial resolution given in the table is the average value among the entire database. . . . .	160

5.19 Micro-expression spotting result in long videos.  $SAMM_{ME}^c$  represents the SAMM cropped-face videos contain ME,  $SAMM_{ME}^f$  are the ME videos with full frame,  $CAS(ME)_{ME}^2$  means all the videos in this sub-dataset of  $CAS(ME)^2$  have ME sequences. . . . . 161

B.1 Summary table for published micro-expression recognition articles with their corresponding metrics and databases . . . . . 178