Micro-Facial Movements: An Investigation on Spatio-Temporal Descriptors

Adrian K. Davison[1](✉), Moi Hoon Yap[1], Nicholas Costen[1], Kevin Tan[1], Cliff Lansley[2], and Daniel Leightley[1]

[1] Manchester Metropolitan University, Manchester M1 5GD, UK
{A.Davison,M.Yap,N.Costen,K.Tan,D.Leightley}@mmu.ac.uk
[2] The Emotional Intelligence Academy, Walkden M28 7BQ, UK
Cliff@eiacademy.co.uk

Abstract. This paper aims to investigate whether micro-facial movement sequences can be distinguished from neutral face sequences. As a micro-facial movement tends to be very quick and subtle, classifying when a movement occurs compared to the face without movement can be a challenging computer vision problem. Using local binary patterns on three orthogonal planes and Gaussian derivatives, local features, when interpreted by machine learning algorithms, can accurately describe when a movement and non-movement occurs. This method can then be applied to help aid humans in detecting when the small movements occur. This also differs from current literature as most only concentrate in emotional expression recognition. Using the CASME II dataset, the results from the investigation of different descriptors have shown a higher accuracy compared to state-of-the-art methods.

Keywords: Micro-movement detection · Facial analysis · Random forests · Support vector machines

1 Introduction

Detecting micro-facial movements (MFMs) is a new and challenging area of research in computer vision that has been inspired by work done by psychologists studying micro-facial expressions (MFEs) [7,12]. Facial expressions have strong scientific evidence suggesting they are universal rather than culturally defined [6]. When an emotional episode is triggered, there is an impulse that cannot be controlled which may induce one of the 7 universal facial expressions (happy, sad, anger, fear, surprise, disgust or contempt). When a person consciously realises that this facial expression is happening, the person may try to suppress the facial expression. Doing this can mask over the original facial expression and cause a transient facial change referred to as a MFE. The speed of these MFEs are high, typically less than 1/5th of a second. During experiments [6] where videos were recorded at 25 frames per second (fps), MFEs have been found to last 1/25th of a second.

L. Agapito et al. (Eds.): ECCV 2014 Workshops, Part II, LNCS 8926, pp. 111–123, 2015.
DOI: 10.1007/978-3-319-16181-5_8

MFEs are not so straightforward that they can be interpreted as an emotion and require the context of when the movement occurred to understand whether the movement can be classed as an MFE or as an MFM. Both can be coded objectively using the Facial Action Coding System (FACS) [5], which defines muscle movements and intensity on the face with no emotional interpretation.

The process of detecting normal facial expressions in computer vision usually involves preprocessing, feature extraction and classification. Methods such as Support Vector Machines (SVM) or Random Forests (RF) [21,26] are used to classify and recognise an emotion. This process is similar for MFEs and MFMs, however the features used must be descriptive enough to detect a movement has occurred, because large movements of normal facial expressions usually have more descriptive features making them easier to detect.

Due to the problems described above, this paper extracts local features from image sequences using local binary patterns on three orthogonal planes (LBP-TOP) and Gaussian derivatives (GDs) to accurately determine that a micro-movement has occurred on a face within the dataset compared with an image sequence where no movement occurs (neutral expression). Using these features, two classifiers, SVM and RF, are investigated in how they classify the movements. From the results, a human interpreter would be able to see any movements they may miss, and it can help in interpreting what the movements may mean in the context of the situation.

The remainder of this paper is divided into the following sections; Section 2 discusses related work and approaches in current literature. Section 3 and 4 describe our investigation of detecting micro-facial movement against a neutral face and the results from experiments respectively. Finally, section 5 concludes this paper.

2 Related Work

Previous work in this field is limited, with current literature focusing on recognising what emotion has occurred, and not when a movement occurs.

Pfister et al. [15] use temporal interpolation with multiple kernel learning and RF classifiers on their own spontaneous micro-expression corpus (SMIC dataset) [11]. The authors classify a MFE into positive or negative categories depending on two annotators labelling based on subjects' self reported emotions. Polikovsky et al. [16] introduce another new dataset recorded at 200 frames per second (fps) and the face images are divided into regions created from manually selected points. Motion in each region is then calculated using a 3D-Gradient orientation histogram descriptor. Shreve et al. [19] propose an automatic method of detecting macro- and micro-expressions in long video sequences by utilising the strain on the facial skin as a facial expression occurs. The magnitude of the strain is calculated using the central difference method over the dense optical flow field observed in regions of the face. Wang et al. [23,24] use discriminant tensor subspace analysis and extreme learning machine as a novel way of recognising faces and MFEs. The authors take a grey-scaled facial image and treat it as a second order tensor and adopt two-sided transformations to reduce dimensionality. Further, they use a tensor independent color space model to show performance of MFE recognition in a different colour space compared with RGB and grey-scale.

Local Binary Pattern (LBP) features [14] form labels for each pixel in an image by thresholding a 3x3 neighbourhood of each pixel with the centre value. The result is a binary number where if the outside pixels are equal to or greater than the centre pixel, it is assigned a 1, otherwise 0. The amount of labels will therefore be $2^8 = 256$ labels. This operator was extended to use neighbourhoods of different sizes [13]. Using a circular neighbourhood and bilinearly interpolating values at non-integer pixel coordinates allow any radius and number of pixels in the neighbourhood. The grey-scale variance of the local neighbourhood can be used as the complementary contrast measure.

As a further extension to local binary patterns (LBP) as a static texture descriptor, Zhao et al. [27] take the LBP in three orthogonal planes, these planes being the spatial and temporal planes (XY, XT ,YT). Originally for dynamic texture recognition, it was used alongside volume LBP to recognise facial expressions. However, unlike dynamic textures, the recognition of facial expressions was done by dividing the facial expression image sequence into blocks and computing the LBP for each block in each plane. These LBP features were then concatenated to form the final LBP-TOP feature histogram. The LBP-TOP histogram provides a robustness to pose and illumination changes, and as the images are split into blocks, the local features of the face better describe facial expressions than a global description of the whole face would.

The Gaussian function is a well-known algorithm and is usually referred to being a normal distribution. Ruiz-Hernandez et al. [18] use the second order derivative to extract blobs, bars and corners to eventually use the features to detect faces in a scene. GDs also provide a powerful feature set with scale and rotation invariant image description. However, when processing higher order derivatives, the feature selection becomes more sensitive to noise, and computationally expensive.

Classification in this area is well established. Random Forests [2] are an ensemble learning method used for classification. It uses many decision trees to find an average balance of votes to decide where a feature should be classified. As a supervised learning method, RF will require training from processed images from a dataset. Support Vector Machines [4] is another supervised learning algorithm which finds the optimal separating hyperplane to decide where to classify data. Both RF and SVM have been used in facial expression recognition [15,21,25,27] and also in other methods such as physical rehabilitation [10] and bioinformatics [20].

The investigation of this paper does not attempt to recognise MFEs as most others, and treats the problem as detecting whether a MFE has occurred compared with a sequence of images that does not contain any movement. These two classes can then be classified using RF and SVM. The potential application of this is to aid a person in detecting when the micro-movement has occurred, and then use this to interpret potential emotion.

3 Method

This section describes a method of differentiating between a MFM and a neutral expression. Normalisation is described by automatically using the centre point of the two eyes and affine transformation to rotate each face from CASME II [25], and then cropping each image to just the face itself. Finally, LBP-TOP and GD features are obtained and classified into either a MFM or neutral expression using RF and SVM.

3.1 Normalisation

Normalisation is applied to all sequences so that all the faces are in the same position based on a constant reference point, in this case, the midpoint between the eyes. Once the midpoint has been obtained, affine transformation is used to rotate the face so that all faces line up horizontally based on this point. The face of the sequences then needs to be cropped to remove the unnecessary background in each image.

To calculate the midpoint of the eyes, first the centre of both eyes are obtained automatically by using a Viola-Jones Haar cascade detector [22] to detect both the left and right eyes separately. Closed eye Haar detectors are available, however as the dataset does not include closed eyes, this has not been implemented. This creates a bounding box around both eyes which the centre point of an eye can then be extracted

$$(C_x, C_y) = \left(\frac{W}{2} + x, \frac{H}{2} + y\right) \tag{1}$$

where C is the centre of the eye, W is the width of the bounding box, and H is the height and x and y are the pixel locations of the top-left corner of the bounding box for the eye. Once the centre points are found for both the left and the right eye, this paper computes the midpoint of the eyes

$$(M_x, M_y) = \left(\frac{LC_x + RC_x}{2}, \frac{LC_y + RC_y}{2}\right) \tag{2}$$

where M is the midpoint between the eyes and LC and RC are the centres of the left and right eye respectively. Using the calculated points, it can be worked out how to apply affine transformation to all images. First the distance between the eyes in Eq. 3 is found and then the angle between the eyes is calculated in Eq. 4

$$(D_x, D_y) = (|RC_x - LC_x|, |RC_y - LC_y|) \tag{3}$$

$$\theta = \frac{\arctan(D_x, D_y)180}{\pi} \tag{4}$$

where D is the distance between the eyes and θ is the angle between the eyes. Using the extracted points, affine transform is used to align the eyes horizontally, ready to be processed.

3.2 Processing Images

Feature extraction begins by grey scaling each image sequence and dividing each image into 9x8 non-overlapping blocks, as proposed by Zhao et al. [27] as their best performing block size (see Fig. 1). This sequence then has a GD operator applied with σ (the standard deviation) being changed from 1-7 in each iteration once the whole database has been processed.

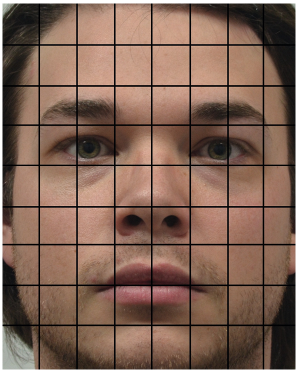

Fig. 1: Images are split into 9x8 blocks so each can be processed separately and obtain local features that are concatenated to form the overall global feature description

A Gaussian function is used as a blurring filter to smooth an image, lowering the high frequencies denoted as

$$G(x, y; \sigma) = e^{-\frac{x^2+y^2}{2\sigma^2}} \tag{5}$$

To extract features such as blobs and corners from the face images, the first and second order derivatives [17] of the Gaussian function is calculated. The first order GD is defined as

$$G_x(x, y; \sigma) = \frac{\partial G(x, y; \sigma)}{\partial x} = -\frac{x}{\sigma^2} G(x, y; \sigma) \tag{6}$$

$$G_y(x, y; \sigma) = \frac{\partial G(x, y; \sigma)}{\partial y} = -\frac{y}{\sigma^2} G(x, y; \sigma) \tag{7}$$

where σ is the scaling element of the GD. The second order GD is defined as

$$G_{xx}(x, y; \sigma) = \left(\frac{x^2}{\sigma^4} - \frac{1}{\sigma^2} \right) G(x, y; \sigma) \tag{8}$$

$$G_{yy}(x, y; \sigma) = \left(\frac{y^2}{\sigma^4} - \frac{1}{\sigma^2} \right) G(x, y; \sigma) \tag{9}$$

$$G_{xy}(x,y;\sigma) = \frac{xy}{\sigma^4} G(x,y;\sigma) \tag{10}$$

the first and second order derivative features are then summed together to get the final GD feature and form a stronger feature representation of blobs, corners and other important features. LBP-TOP is then applied as follows: each block has the standard LBP operator applied [13] with α being the centre pixel and P being neighbouring pixels with a radius of R

$$LBP_{P,R} = \sum_{p=0}^{P-1} s(g_p - g_\alpha) 2^p \tag{11}$$

where g_α is the grey value of the centre pixel and g_p is the grey value of the p-th neighbouring pixel around R. 2^p defines weights to neighbouring pixels and is used to convert the binary string pattern into a decimal. The sign function to determine the binary values assigned to the pattern is

$$s(x) = \{^{1, x \geq 0}_{0, x < 0} \tag{12}$$

where if x is greater than or equal to 0 then $s(x)$ is 1, otherwise 0. After the image has been assigned LBPs, the histogram can be calculated

$$H_i = \sum_{x,y} I\{f_l(x,y) = i\}, i = 0, ..., n-1 \tag{13}$$

where $f_l(x,y)$ is the image labelled with LBPs. Completing this task on only the XY plane would be suitable for static images, however calculating the XY, XT and YT planes is required to gain a spatio-temporal view of the sequence of images, as expressions are much better described in the temporal domain than still frames [1]. Each plane has been divided into blocks and the LBP histograms extracted to be concatenated into the final feature histogram to be used in classification. For this method, the radius R was set to 3 and the neighbouring points P was set to 8. Fig. 2 shows a representation of creating the LBP-TOP features.

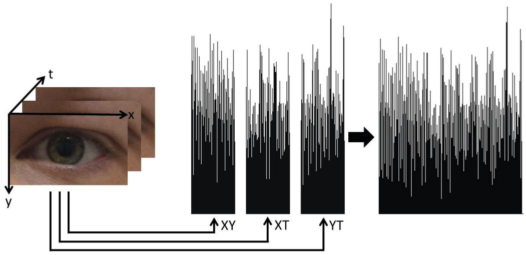

Fig. 2: LBP is calculated on every block in all three planes. Each plane is then concatenated to obtain the final LBP-TOP feature histogram

3.3 Classification

Two popular data classification methods, SVM and RF, will be used to classify between micro-movement and neutral faces within the whole of the CASME dataset. The results of the experiment will compare MFMs against neutral face sequences.

A RF model is constructed by using the bootstrap method to randomly generate a number of decision trees (n_{tree}), which are each provided with randomly selected samples of the training input and then all decision trees are combined into a decision forest. For each bootstrap, a random sample of the training data is used which determines the size of an un-pruned classification tree (*mtry*, default 3). Voting from all trees is used for classification, with the highest voted choice within the data to be selected.

The selected data is taken from the CASME II dataset and consists of micro-movement and neutral face sequences. The RF will determine the accuracy based on correctly classified labels against incorrectly classified labels. RF requires one parameter, n_{tree}, which sets the number of trees to grow. In this experiment the number of trees was set to 300. The software used to implement RF was randomForest Toolbox for Matlab [9].

SVM attempts to find a linear decision surface (hyperplane) that can separate the two classes and has the largest distance between support vectors (elements in data closest to each other across classes). If a linear surface does not exist, then the SVM maps the data into a higher dimensional space where a decision surface can be found. The kernel selected for SVM is the radial basis function (RBF) and will use the same movement and neutral data as RF to determine the accuracy based on the correctly classified labels against incorrectly classified labels.

There are two main parameters that will be selected: Parameter c is a user-defined parameter that controls the trade-off between model complexity and empirical error in SVM. In addition, the parameter γ determines the shape of the separating hyperplane in the RBF kernel. Selection of the optimised parameters was undertaken according to the method by Hsu et al. [8]. The classifier was trained on one subset (training data) and accuracy is tested with the introduction of the second subset (testing). The optimisation process was repeated for each of the possible parameter in exponential steps for both c and γ between 2^{-10} to 2^{10} and 2^{-3} to 2^{3} respectively. The software used to implement SVM is libSVM Toolbox for Matlab [3].

4 Experimental Results

To test this method's performance, combinations of image planes are used with temporal and spatial mixes. The testing data is set up to 50%, therefore if 30% is training the remaining 70% is used for testing. No data within the training set is used for testing to ensure all testing data is unseen. Each plane is tested using 100-fold cross-validation. Other literature [15,25,27] use leave-one-subject-out evaluation with data. This paper uses more or equal testing than training

Table 1: All results using the SVM classifier. Each plane used the the combination of LBP-TOP and GD features. The training percentage is displayed for each plane from 10% to 50%.

Plane	σ	**10% Train.**	**20% Train.**	**30% Train.**	**40% Train.**	**50% Train.**
		Accuracy (%)	**Accuracy (%)**	**Accuracy (%)**	**Accuracy (%)**	**Accuracy (%)**
XT	1	51.20	47.10	43.30	39.90	36.20
XY	1	51.10	46.90	43.10	39.10	35.10
XTYT	1	51.90	47.90	44.20	40.00	36.80
YT	1	51.00	46.90	43.10	39.40	35.80
All Planes	1	52.50	48.70	44.80	40.90	37.50
XT	2	52.40	48.80	44.80	41.10	36.90
XY	2	52.10	48.10	44.30	40.20	36.40
XTYT	2	53.20	49.80	46.80	43.60	40.90
YT	2	52.50	48.80	45.10	41.30	37.70
All Planes	2	53.70	51.30	48.80	46.30	43.90
XT	3	52.30	48.50	44.60	41.00	37.20
XY	3	52.30	48.30	44.50	40.70	37.10
XTYT	3	53.20	50.20	47.50	44.70	41.70
YT	3	52.60	48.70	45.50	41.60	38.50
All Planes	3	54.20	52.00	50.10	48.30	46.20
XT	4	52.30	48.20	44.40	40.90	36.90
XY	4	52.40	48.30	44.60	40.70	37.10
XTYT	4	53.30	50.20	47.40	44.30	41.20
YT	4	52.40	48.70	45.30	41.90	38.50
All Planes	4	***54.30***	52.30	50.20	48.40	46.60
XT	5	49.82	46.33	42.47	39.56	36.21
XY	5	50.62	46.45	42.74	39.12	35.65
XTYT	5	51.51	47.36	44.03	40.18	37.00
YT	5	50.00	46.12	42.94	40.20	36.56
All Planes	5	52.28	48.26	44.32	40.62	36.40
XT	6	49.89	45.64	42.59	39.03	35.70
XY	6	50.47	46.04	42.47	38.66	35.02
XTYT	6	51.08	47.17	43.64	39.78	36.37
YT	6	49.84	45.86	42.38	38.95	36.11
All Planes	6	52.24	48.08	44.02	39.91	36.26
XT	7	49.62	45.40	41.87	38.27	34.93
XY	7	50.30	46.02	42.26	38.47	34.73
XTYT	7	50.81	46.79	43.36	39.43	36.13
YT	7	49.75	45.87	42.53	39.25	36.43
All Planes	7	52.14	48.01	43.91	39.83	36.09

to describe the robustness of this method compared to others in the literature. The dataset being used is the CASME II recorded at 200 fps with 35 Chinese participants with a mean age of 22.03 years.

For both RF and SVM the σ value for GDs goes from $1 - 7$. In RF the accuracy increases until the 5th value, where it peaks and begins to decrease, indicating that when $\sigma = 5$ the accuracy is at its highest. In SVM, the accuracy decreases as the σ value increases.

Table 1 shows the results from the SVM experiment and Table 2 shows results from the RF experiment. SVM and RF results vary considerably with the highest accuracy for SVM was 54.3% with training set to 10%. The accuracy

Table 2: All results using the RF classifier. Each plane used the the combination of LBP-TOP and GD features. The training percentage is displayed for each plane from 10% to 50%. The results for RF are significantly higher than SVM with results starting to plateau and decrease when $\sigma = 6$.

Plane	σ	**10% Train. Accuracy (%)**	**20% Train. Accuracy (%)**	**30% Train. Accuracy (%)**	**40% Train. Accuracy (%)**	**50% Train. Accuracy (%)**
XT	1	59.00	63.00	65.60	67.00	70.60
XY	1	51.20	49.30	48.10	46.30	44.50
XTYT	1	57.60	61.20	63.80	66.00	68.00
YT	1	56.60	59.00	60.50	62.70	65.30
All Planes	1	55.80	57.60	58.60	60.00	60.70
XT	2	66.80	73.70	77.20	80.70	82.30
XY	2	51.80	49.80	48.20	45.60	43.90
XTYT	2	66.10	72.20	75.90	79.30	81.60
YT	2	64.90	70.70	75.00	77.70	80.30
All Planes	2	61.30	66.40	69.50	71.20	74.00
XT	3	74.30	82.80	85.90	88.50	90.10
XY	3	52.90	50.80	49.40	48.30	46.20
XTYT	3	73.90	81.80	85.40	87.90	89.20
YT	3	72.30	80.20	84.30	86.50	88.00
All Planes	3	68.10	74.70	78.60	81.30	83.80
XT	4	79.40	86.80	89.20	91.30	92.40
XY	4	53.10	51.70	50.10	48.40	46.60
XTYT	4	78.50	86.10	88.50	90.90	91.70
YT	4	77.80	84.60	87.40	89.00	90.80
All Planes	4	70.60	78.10	81.70	84.80	86.70
XT	5	78.80	86.50	89.50	91.20	92.50
XY	5	53.30	51.50	49.80	47.10	45.40
XTYT	5	79.30	86.70	89.20	91.40	***92.60***
YT	5	78.30	85.70	88.70	90.60	92.20
All Planes	5	71.70	79.00	82.50	84.80	87.30
XT	6	78.60	85.70	88.70	90.80	91.80
XY	6	52.80	50.60	48.30	46.90	44.50
XTYT	6	78.30	86.10	88.70	90.90	92.00
YT	6	78.40	84.90	87.60	90.00	91.40
All Planes	6	70.30	77.30	80.40	84.30	86.40
XT	7	75.40	83.00	85.90	88.40	89.80
XY	7	52.70	50.20	48.30	45.40	42.80
XTYT	7	77.40	83.90	87.10	89.10	90.60
YT	7	77.60	83.90	87.20	88.80	90.20
All Planes	7	69.20	75.60	79.00	81.00	84.00

gradually decreased as training increased. As the data is high-dimensional and values lie close together, SVM struggles to separate the data beyond chance. As RF uses a bootstrap method it is able to generate many classifiers (ensemble learning) and aggregate results to handle the data more appropriately, only ever choosing random samples and ignoring irrelevant descriptors. This gave the highest accuracy of **92.6%** in the XTYT plane with a standard deviation (STD) of 1.78.

By removing the spatial information and just using the temporal planes, classification results for RF are higher. In SVM the results did not vary considerably

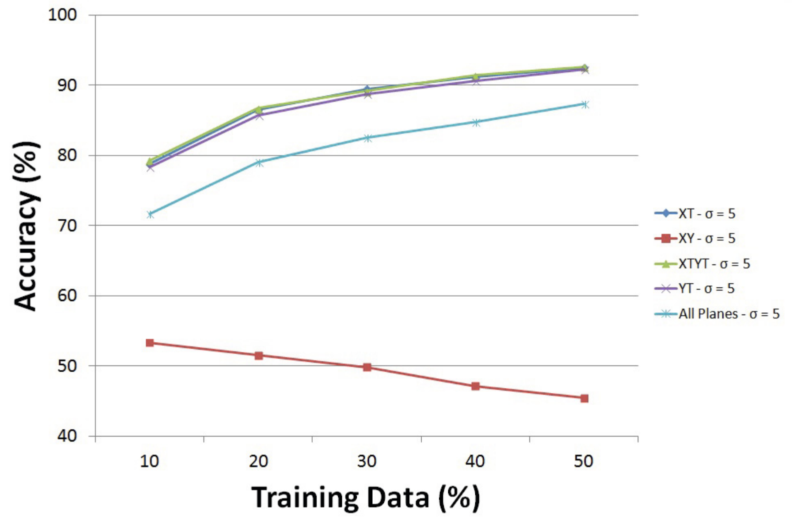

Fig. 3: Using RF, the accuracy of all planes where $\sigma = 5$. Notice XY decreases as training increases due to the lack of temporal information.

Table 3: All results using the SVM classifier when using only LBP-TOP features. The training percentage is displayed for each plane from 10% to 50%.

Plane	**10% Train. Accuracy (%)**	**20% Train. Accuracy (%)**	**30% Train. Accuracy (%)**	**40% Train. Accuracy (%)**	**50% Train. Accuracy (%)**
XT	52.4	48.8	46.1	43	40.9
XY	51.9	48.1	43.9	40.1	36.7
XTYT	53.6	51.1	48.8	46.7	44.2
YT	53.1	50.6	47.9	45	43.1
All Planes	54.2	52.2	50.2	48.3	46.3

across planes, and the highest result was for all planes (54.3%, STD: 0.56) and the lowest being the XY plane alone (34.73%, STD: 2.6).

The highest results were found to be when the σ value was set to 5. Fig. 3 shows the gradual increase in accuracy as training is increased in all planes with a temporal element. A decrease was shown in just the XY plane, supporting that as more training is introduced, the XY plane acts as noise to any movement. This can also be seen when all planes are used and the accuracy is pushed lower than just the temporal planes.

SVM and RF were also used to classify the image sequences using only LBP-TOP features. The results in Table 3 show that all of the planes perform no much better than chance, if not lower, with accuracy decreasing as the amount of training data is introduced. SVM appears to perform similar to results with GD, and separating the features is difficult. Table 4 shows the results from RF using only LBP-TOP features. The accuracy for detecting movement increased significantly compared with SVM, however the highest result was lower than

Table 4: All results using the RF classifier when using only LBP-TOP features. The training percentage is displayed for each plane from 10% to 50%.

Plane	**10% Train. Accuracy (%)**	**20% Train. Accuracy (%)**	**30% Train. Accuracy (%)**	**40% Train. Accuracy (%)**	**50% Train. Accuracy (%)**
XT	60.4	64.3	66.6	69.4	71.4
XY	50.9	48.1	46	43.3	41.4
XTYT	58.8	61.7	63.3	65.5	67.8
YT	56.8	58.7	60.6	62.3	64
All Planes	56	57.8	58.5	59.6	59.9

when combined with GDs at **71.4%** when using 50% training and 50% testing data in the XT plane.

To the best of our knowledge, there has not been any results from purely detecting MFM when comparing with neutral faces and so a benchmark for comparing our results could not be found. Most previous work focuses on detecting the movements and classifying into distinct emotional categories and therefore include automatic interpretation based on the FACS equivalent muscle movements (i.e. happy would be movement in AU12).

5 Conclusion

This paper shows that the combination of LBP-TOP and GD features, classification with RF can perform significantly better than SVM when detection micro-movement against neutral, with a highest accuracy of 92.6%. The standard deviation of results is low indicating mean accuracies are consistent using cross-validation. This paper also shows that combining the higher order GD and LBP-TOP can represent the subtle temporal movement of the face well with RF. However, the features are unable to be split by the SVM hyperplane beyond chance.

When using spatial XY planes alone or combined with temporal planes, detection accuracy decreases, suggesting the XY plane is introducing noise to subtle movement. Our method specifically detects micro-movement against neutral faces, which has yet to become a well established method. Most current research detects the MFEs to then classify them into emotion categories.

Future work will look into how the data is represented for MFMs and MFEs, including exploring further methods of temporal feature selection and extraction for micro-movements and how best to discriminate clearly when a subtle movement occurs. Other work includes exploring unsupervised learning methods of classifying movement and non-movement instead of using supervised and computationally expensive methods that require training.

References

1. Bassili, J.N.: Emotion recognition: the role of facial movement and the relative importance of upper and lower areas of the face. Journal of Personality and Social Psychology **37**(11), 2049 (1979)

2. Breiman, L.: Random forests. Machine Learning **45**, 5–32 (2001)
3. Chang, C.C., Lin, C.J.: LIBSVM: A library for support vector machines. ACM Transactions on Intelligent Systems and Technology **2**, 27:1–27:27 (2011). http://www.csie.ntu.edu.tw/~cjlin/libsvm
4. Cortes, C., Vapnik, V.: Support-vector networks. Machine Learning **20**, 273–297 (1995)
5. Ekman, P., Friesen, W.: Facial Action Coding System: A Technique for the Measurement of Facial Movement. Consulting Psychologists Press (1978)
6. Ekman, P.: Emotions Revealed: Understanding Faces and Feelings. Phoenix (2004)
7. Ekman, P.: Lie catching and microexpressions. In: The Philosophy of Deception. Oxford University Press (2009)
8. Hsu, C.W., Chang, C.C., Lin, C.J., et al.: A practical guide to support vector classification (2003). https://www.cs.sfu.ca/people/Faculty/teaching/726/spring11/svmguide.pdf
9. Jaiantilal, A.: Random forest (regression, classification and clustering) implementation for matlab (2009). http://code.google.com/p/randomforest-matlab
10. Leightley, D., Darby, J., Li, B., McPhee, J., Yap, M.H.: Human activity recognition for physical rehabilitation. In: International Conference on Systems, Man, and Cybernetics (SMC), pp. 261–266 (2013)
11. Li, X., Pfister, T., Huang, X., Zhao, G., Pietikäinen, M.: A spontaneous micro-expression database: Inducement, collection and baseline. In: FG (2013)
12. Matsumoto, D., Hwang, H.S.: Evidence for training the ability to read micro-expressions of emotion. Motivation and Emotion **35**, 181–191 (2011)
13. Ojala, T., Pietikainen, M., Maenpaa, T.: Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. PAMI **24**, 971–987 (2002)
14. Ojala, T., Pietikainen, M., Harwood, D.: A comparative study of texture measures with classification based on featured distributions. Pattern Recognition **29**(1), 51–59 (1996)
15. Pfister, T., Li, X., Zhao, G., Pietikainen, M.: Recognising spontaneous facial micro-expressions. In: ICCV (2011)
16. Polikovsky, S., Kameda, Y., Ohta, Y.: Facial micro-expressions recognition using high speed camera and 3d-gradient descriptor. In: ICDP (2009)
17. Romeny, B.M.H.: Gaussian derivatives. In: Front-End Vision and Multi-Scale Image Analysis. Springer, The Netherlands (2003)
18. Ruiz-Hernandez, J., Lux, A., Crowley, J.: Face detection by cascade of gaussian derivates classifiers calculated with a half-octave pyramid. In: International Conference on Automatic Face Gesture Recognition, pp. 1–6, September 2008
19. Shreve, M., Godavarthy, S., Goldgof, D., Sarkar, S.: Macro- and micro-expression spotting in long videos using spatio-temporal strain. In: FG (2011)
20. Statnikov, A., Wang, L., Aliferis, C.: A comprehensive comparison of random forests and support vector machines for microarray-based cancer classification. BMC Bioinformatics **9**(1), 319 (2008)
21. Tian, Y.L., Kanade, T., Cohn, J.F.: Facial expression analysis. In: Handbook of Face Recognition. Springer, New York (2005)
22. Viola, P., Jones, M.: Rapid object detection using a boosted cascade of simple features. In: CVPR (2001)
23. Wang, S.J., Chen, H.L., Yan, W.J., Chen, Y.H., Fu, X.: Face recognition and micro-expression recognition based on discriminant tensor subspace analysis plus extreme learning machine. Neural Processing Letters **39**, 25–43 (2014)

24. Wang, S.J., Yan, W.J., Li, X., Zhao, G., Fu, X.: Micro-expression recognition using dynamic textures on tensor independent color space. In: ICPR (2014)
25. Yan, W.J., Li, X., Wang, S.J., Zhao, G., Liu, Y.J., Chen, Y.H., Fu, X.: Casme ii: An improved spontaneous micro-expression database and the baseline evaluation. PLoS ONE **9**, e86041 (2014)
26. Zeng, Z., Pantic, M., Roisman, G., Huang, T.: A survey of affect recognition methods: Audio, visual, and spontaneous expressions. PAMI **31**, 39–58 (2009)
27. Zhao, G., Pietikainen, M.: Dynamic texture recognition using local binary patterns with an application to facial expressions. PAMI **29**, 915–928 (2007)