Emotion Recognition Through Facial Expressions

Xavier Walcome
Division of Science and Mathematics
University of Minnesota, Morris
Morris, Minnesota, USA 56267
walco005@morris.umn.edu

ABSTRACT

This paper discusses modern styles of emotion recognition, including local binary patterns from three orthogonal planes and support vector machines, though several others are discussed to a lesser degree, just not to the extent of these. Discussion of how all of these compare and why they perform how they do is also included.

Keywords

Emotion Recognition, LBP-TOP, Support Vector Machines

1. INTRODUCTION

Recognizing emotion, for most of us, is something we do not even think about. It's almost instantaneous when we look at a persons face or hear a certain tone in their voice. For a computer, though, this is a very taxing challenge. It would be comparable to a person who knows nothing about flowers to identify a specific flower. There's a large prerequisite of knowledge needed for a computer to identify what emotion a face is showing, like how a botanist needs training to be able to tell what the type of a certain flower is. Emotion recognition has been a very relevant issue in human-computer interaction, being able to have a computer predict a users emotion has a plethora of applications, such as detecting depression or assisting people who cannot recognize emotion, such as people with autism.

There is an annual challenge done through ACM, a collection of conferences related to computer science, called the Emotion Recognition in the Wild Challenge, or emotiW. Each year they give a set of data in the form of a couple thousand video clips and certain baselines to teams that sign up, then each team creates a system to recognize the emotion from these video clips. The 2014 emotiW challenge is where two of the three studies that will be discussed are taken from, which gave the teams two thousand clips to predict emotion from.[2] Most of these use audio and visual features, but for our case we will only cover the visual aspects of their systems and use results exclusively from those if possible.

The local binary patterns from three orthogonal planes operator (LBP-TOP), introduced in 2007 [10], is a very popular way of getting information from a video of a face, it is used in a majority of the submissions of the emotiW challenges. Used along with support vector machines, this can be a very effective way of predicting and recognizing emotion. I will discuss how recent studies compare to each other, more specifically ones that use the LBP-TOP operator against others that do not.

UMM CSci Senior Seminar Conference, December 2015 Morris, MN.

2. FACIAL INFORMATION

The process of recognizing and predicting what emotion a face is conveying begins with converting a series of frames from a video of the face to information that the emotion can be extracted from.

2.1 Local Binary Patterns

Though local binary patterns (LBP) only applies to a single image, the operator that will be discussed builds off of it and understanding of LBP is essential to understanding LBP-TOP. is a process of weighing the gray scale levels of the pixels to neighboring areas. LBP can be used to locate where edges or other contours are in an image. A monochrome image is required for this because it uses gray scale levels of pixels to compare the lighting of the pixels. LBP produces information in the form of a histogram that contains how often each gray scale value shows up in an image.[6]

The LBP operator begins by being given a radius and an amount of sampling points to use. Usually the amount of sampling points, s, is anywhere from eight to 16 and the radius, r, is one to four pixels. First, the image is split into square cells, the size of which is based on the radius. A circle of radius r with its origin at the center pixel should fit perfectly in the cell. The image is split into cells because using the LBP operator on each of these cells, then concatenating the results together stores information about the location of darker or lighter pixels. The LBP operator is then performed on each of these cells. The circumference of the circle with r radius around this center pixel of the cell is called the neighborhood around the center pixel. Then the gray scale levels of s points in the neighborhood are recorded and compared to the gray scale level of the center pixel. If one points gray scale level is less than that of the center pixel, it is darker and denoted with a 0, if not, it is lighter and denoted with a 1. This produces the LBP code

for that pixel, a binary string with s digits. This can then be converted to a decimal, as shown in Figure 1. Using more sampling points allows for more accuracy but increases the processing power needed. A larger radius for the neighborhood can lead to more accuracy, but only when the amount of sampling points is scaled up with it. This might not seem like an issue when applied to just one image, but the amount of computing needed builds up when applied to a video with a high frame rate.

The last step of LBP is converting the LBP codes of each pixel into a histogram. Since the LBP code is a binary value, the amount of labels in each histogram is 256, the amount of possible gray scale values. It looks like a bar graph, but the y axis contains how often each gray scale value occurs in the cell. These histograms are then concatenated together, from top left to bottom right. So the final histogram starts with the first histogram on the top left, then the next column after the 256th column is the first column of the next histogram. This is done so the location of A basic example of the histogram is shown in section c of Figure 4. [6]

2.2 Local Binary Patterns From Three Orthogonal Planes

Local Binary Patterns From Three Orthogonal Planes (LBP-TOP) is an extension of LBP, but as opposed to how LBP is used on a single image, LBP-TOP is used on a series of frames. The three planes mentioned in the name are: XY, a single frame; XT, the plane showing how a row changes over time; YT, the plane showing how a column changes over time. The X and Y values are the X and Y axes of one sample frame from the video and the T value is based on the amount of frames used. These three planes intersect on a center pixel, as shown in Figure 2. The amount of neighboring points and the distance of the radius can differ on each of these planes, because these different planes dimensions may vary. Figure 2 shows this in the example of the planes to the right. These will all be done on a case-by-case basis because resolution and the amount of frames per second affects how relevant a smaller radius would be versus a larger radius. The radius of any of these planes aren't dependent on one another. For example, a video that has a slow frame rate would have a lower radius on the XT and YT planes and a higher frame rate would have a higher radius on these planes. Also, the radii are most likely to be ellipses for the XT/YT planes, because the time portion of the plane is probably longer than the X or Y portion. This is shown in Figure 3, where the radii are all different for each plane.

[10] Like how LBP is done by splitting the image into cells, this set of frames is then split into overlapping blocks on the XY frame and then LBP-TOP is performed on each of these and the histograms produced are concatenated. Figure 4 shows this process. Splitting the frames up in this way, as it does in LBP, stores the spacial information of each section of the face, something very important for extracting emotion from a face. Overlapping blocks also allow for more accuracy because the radii used when LBP is done on each of these blocks may miss some corners of these blocks.

3. EMOTION PREDICTION

After pre-processing has occurred on a video and puts it into a certain set of data, this data can be used to predict emotion. Most of the recent studies done on emotion recognition use some form of machine learning.

Machine learning is a process by which you train a model on a set of data, giving it person-reviewed data as well as what certain category the data is placed in. In this case, what you give a model to recognize emotion is whatever data the facial recognition process gave you and what emotion it conveys. Usually data sets for machine learning need to be very large to allow for better accuracy.

After the model is trained, you can give it a new set of data and it will put it in the category it believes the data is based on how similar it is to other data already in the model. Once that data is processed and checked by humans to make sure the labels given by the machine are correct, it is then put into the model and used as a comparison for new data. Thus, the model learns as it is given data to process.

This is just a very basic intro to this, since the complexity of machine learning is beyond the scope of this paper. The main type of machine learning that I will focus on is the support vector machine, since it is used in most of the emotiW challenges that will be covered later in the paper.

3.1 Support Vector Machines

Support vector machines (SVMs) are used very often in emotion recognition because they are very efficient when it comes to dealing with high-dimensional data like emotion. Each dimension is a different emotion, including neutral, so that means that if all the emotional data is compared in one SVM it would have seven dimensions, one for each base emotion: happiness, anger, sadness, surprise, fear, disgust, and a neutral emotion. Each data point in an SVM needs to have a *dot product*, or *inner product*, which is a way for *vectors* in a data set to be compared to each other. Vectors are a one-dimensional collection of data, comparable to a list. In the case of emotion recognition, each vector would contain the concatenated histogram that LBP-TOP produced.

SVMs can also compare data in another way without having to deal with using spaces of such a high dimension by using *one-versus-one* and *one-versus-many* comparisons. In this case, each emotion would be a label instead of a dimension. In one-versus-one comparison, one of the labels is compared to another, i.e. sadness and fear. In one-versus-many, there are only two labels as well: data that has a certain label and data that does not have this label, i.e. sadness and not sadness. Using these allows for only two labels to be used per SVM and also makes it easier to understand the data that is being worked with.

The support vector machine, or SVM, creates a *hyperplane*, or *decision boundary* a possibly multi-dimensional plane dividing the data into categories, as well as a margin on either side of the hyperplane that goes from the closest point on one side of the hyperplane to the closest data point on the other side. There can be many different hyperplanes that divide the current data set between its two labels, but the ideal hyperplane is one that has a maximized margin, since this allows for future data to be more accurately predicted. An example of an ideal margin compared to a non-ideal margin is shown in This maximization is assisted by using certain *support vectors*, data points that are closest to points that have other labels. [1]

3.2 Two-Class, Two-Dimensional SVMs

To understand this concept more easily, I will go over a

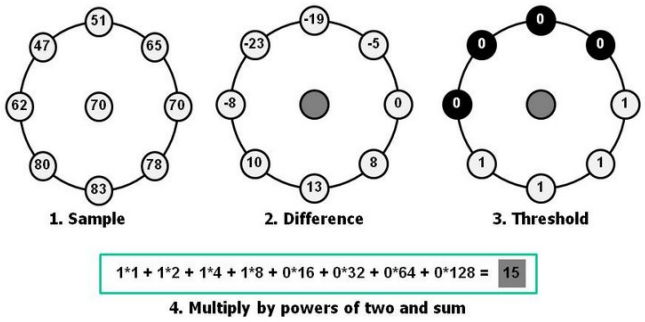

Figure 1: The process to get the LBP code of a pixel[6]

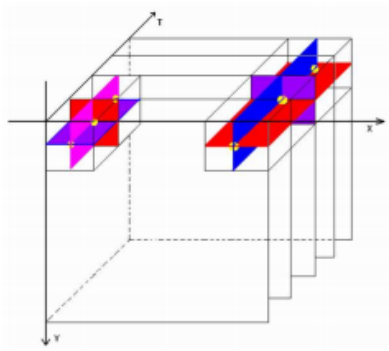

Figure 2: The XY, XT, and YT planes for LBP-TOP[10]

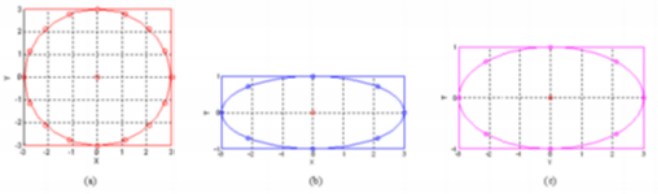

Figure 3: The radii for an example XY, XT, and YT planes respectively[10]

basic two-class *linear classifier*. This is a classifier that does not have to warp the data in any way to provide a decision boundary. The two labels used are shown in the data set in Figure 6.

$$f(x) = w^T x + b \tag{1}$$

Formula 1 shows the *discriminant function* that defines the decision boundary. The discriminant function decides which side of the decision boundary a data point will be on. w is the weight vector, an arbitrary line that, translated to the right **b** units separates the positive data from the negative data.

$$w^T x$$

The above portion of the equation above forms a perpendicular line going out of w in the direction of the point x and defines the distance a data point is away from the decision boundary. The function:

$$w^T x + b = 0$$

shows any points that are on the hyperplane. The sign of the value returned by f(x) is what decides which side of the hyperplane the data point x is since that shows the direction on the plane that x is from w. An example of this is shown in Figure 6, where values that, when put into the discriminant function, return a negative value are in the red circle label. Now that I've gone over the concept of an SVM on a two-dimensional plane where the data can be separated by a straight line, or is *linearly separable*, we can discuss situations where the data is a bit less simply divided.

3.3 Non-Linearly Separable SVM

An example of a non-linearly separable SVM is shown in Figure 7. When a linear classifier like Equation 1 is used on this data set it will not work well at all since there is no way a straight line can be drawn through the data to accurately separate the different labels. There is a novel solution to this, and that is transforming the two-dimensional plane where the data lays into a three-dimensional space in a way where the data would then be able to be separated by a two-dimensional hyperplane as opposed to the one-dimensional line used in the previous section. This concept begins by defining a function phi, where phi takes a data point d and transforms it into a three-dimensional point in this case, but it can be turned into a multi-dimensional point. As shown in Figure 8 this can be used to find a hyperplane that divides the data points reasonably. Then the spaces where the

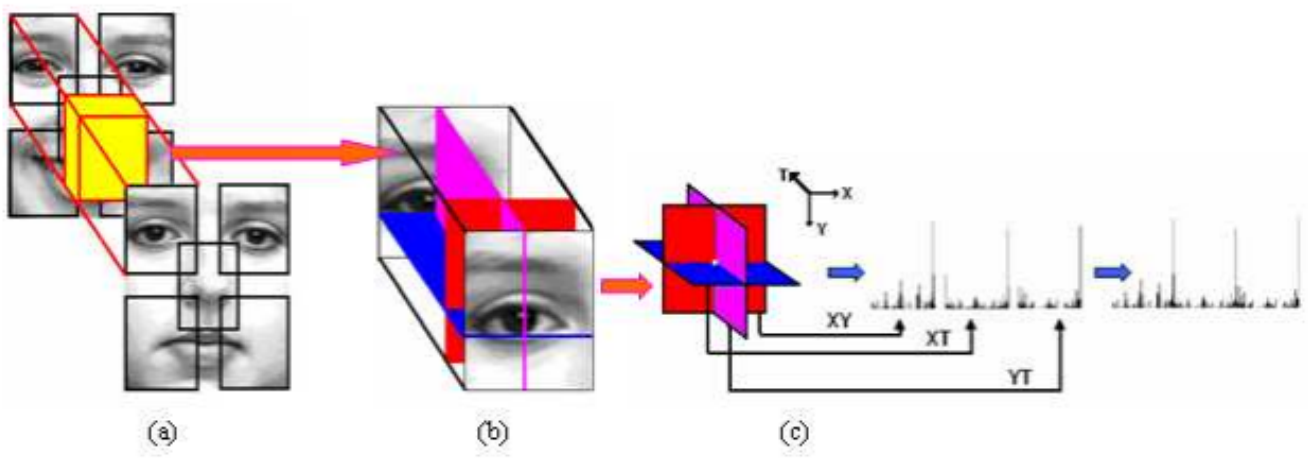

Figure 4: (a) shows an image separated into overlapping blocks. (b) shows LBP-TOP being performed on the set of frames in a certain block. (c) shows the histograms from each plane being concatenated together[10]

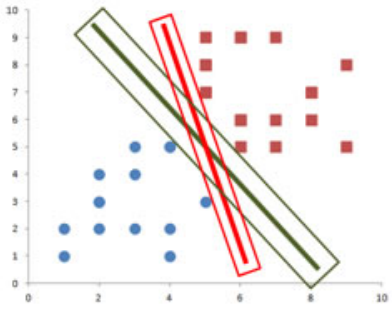

Figure 5: An ideal decision boundary with a maximized margin (green) compared to a non-ideal margin (red)[9].

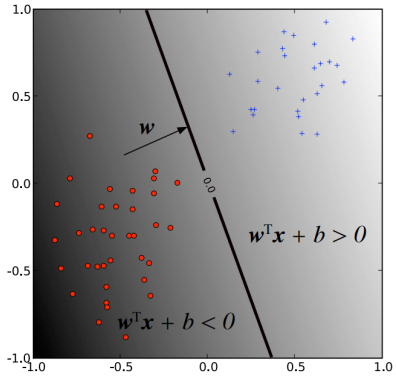

Figure 6: A linearly separable data set divided with a decision boundary decided by an SVM[1].

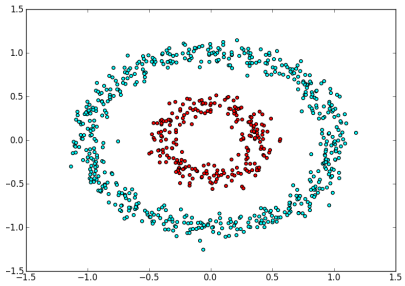

Figure 7: A data set that is non-linearly separable[3].

hyperplane cuts through the "lifted" plane is the non-linear hyperplane on the lower dimensional space.

If the whole data set were converted to a higher dimensional plane then back, that would cause a lot of issues with memory,considering six or seven dimensional planes, those of which are used sometimes in emotion recognition papers that have all the labels compared at once instead of a one-versus-all and one-versus-one strategy that will be discussed later.

3.4 Kernel Functions

To get around this issue you can use *kernel functions*, a way where a data set can compare the dot products of the data points as if they were in a higher dimension. Since each data point has a dot product when it is in its current dimension as well as its higher dimension, you can simply use the dot products of the higher dimension versions of these data points without converting the entire plane to a higher dimension. This saves on a lot of memory because of this and allows for slightly faster computations. There are three kernel functions used: Polynomial kernel, Radial Basis Function (RBF) kernel, and Sigmoid Kernel, shown in Equations 2, 3 and 4 respectively.

Equation 2 shows the polynomial kernel method, with d being the degree specified. A higher degree will yield more precise data, but as it goes up the memory needed also in-

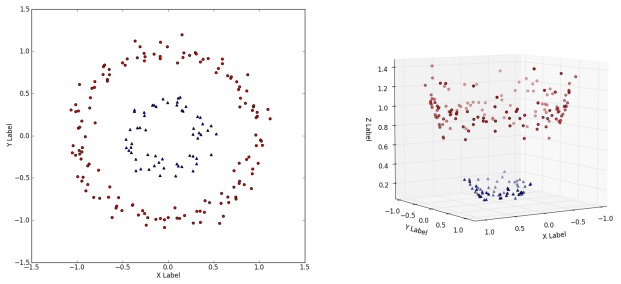

Figure 8: To the right is the data set from Figure 7 in a 2D plane, the left is the same data set but transformed into a 3D plane[3].

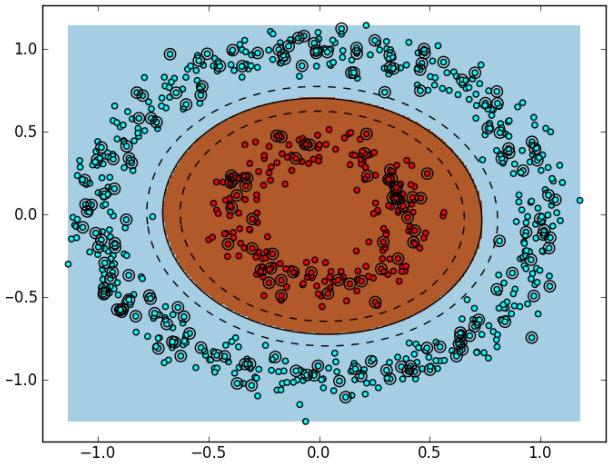

Figure 9: A non-linearly separable data set where a decision boundary has been found using a polynomial kernel[3].

creases. Equation 3 shows the RBF kernel, which produces more of a curve for a decision boundary. Equation 4 shows the sigmoid kernel.

Different problems need the application of different kernels, there is not one kernel that works well with everything. The type of data that is being used is an important factor of deciding which kernel to use. Figure 9 shows the sklearn implementation of the polynomial kernel on the circular data set similar to the one used previously.

$$(\gamma[x, x'] + r)^d \tag{2}$$

$$exp(-\gamma|x - x'|^2) \tag{3}$$

$$tanh(\gamma[x, x'] + r) \tag{4}$$

4. EMOTION RECOGNITION STUDIES

All of these studies are taken from the previously mentioned Emotion Recognition in the Wild challenge, run through ACM annually since 2013. EmotiW provides data sets for these challenges in the form of a couple hundred videos, each labeled with a certain emotion. Since these use video and audio features, when possible, I will discuss only the results from the relevant features from video. When there are no separate results available, the full results will be used.

In the first paper, a study from the Emotion recognition in the wild 2014 challenge, Ringeval et. al. takes an interesting tactic with using the visual data by doing LBP-TOP and an SVM, but also separately using lip activity in a video. [7] They use the LBP-TOP and SVM as discussed, but they also put points on each frame of the face, focusing mainly on the lips because it is believed that the two areas of the face where emotion is mostly conveyed is through the eyes and the lips. They first detected the face then did feature extraction with LBP-TOP. They used the one-versus-one and one-versus-many technique discussed in Section 3.1. They trained these SVMs with the 2832 parameters they had gathered and compared to the baseline of 31.49% accuracy in predicting the correct emotion from the visual information for the EmotiW14 challenge, the accuracy of theirs was 36.13% using a gaussian kernel method. They use multiple two-class SVMs of two different types: one-versus-all, where a data point is compared against one emotion or the rest of the emotions and one-versus-one, where it is compared against two emotions.

In an independently done study, Tarun Krishna et. al. [4] use the process of optic flow in the second paper, where they mark certain points on a face in a set of frames and predict what emotion the person is displaying by calculating the distance these points move. Before this though, they use gabor filtering on the frames, a process that decomposes the image to a black and white image that is simple and removes excess information. After this, the points are applied to the frames. Most of the 66 points they have are around the eyes and the mouth because they are the facial features that have the most information about a faces emotion. They then calculate the distance and direction the points had moved from one frame to another.

They then trained an SVM classifier to classify the data and did a bit lower than the baseline score of 27%, getting only 20%. While they exceeded in detecting happiness and sadness, where their results are better than the baseline percent for these emotions by 28 and 41 percent, the SVM is poor at recognizing neutral expressions and anger, their percentages are less than the baseline by 56 and 54 percent. The results used are from the overall system, since Tarun Krishna et. al. did not release results for just the results of the visual portion of their system. This is denoted by the asterisk in Table 1.

The final paper, an entry for the EmotiW 2014 challenge submitted by Sun et. al. [8] uses LBP-TOP as well as LPQ-TOP to extract facial features and SVMs to train and predict emotion from the data. After detecting and aligning the face, each sequence of frames is divided into four by four blocks, which LBP-TOP is performed on. LPQ-TOP is an alternative version of LBP-TOP which is more robust to blur and is performed on the data separately. Describing this is not part of the scope of the paper, but for more information on LPQ, refer to the paper written by Paivarinta et. al [5]. They used the one-versus-all and the one-versus-one approach as mentioned above as well. When the features are extracted using LBP-TOP the accuracy is 36.12% and when they are extracted using LPQ-TOP it is 19.68%. Both of them used SVMs with a radial basis function kernel method, the only variance is the feature extraction process used. Also, both of these percentages only take into account

Table 1: Results of Studies

Paper	Accuracy	Comments
Ringeval et. al.	36.13%	EmotiW 2014
Krishna et. al.	20.51%	Independent*
Sun et. al. LBP-TOP	36.12%	EmotiW 2014
Sun et. al. LPQ-TOP	19.68%	EmotiW 2014

the video used.

4.1 Results and Discussion

Table 1 shows the results of all the papers discussed in the previous section. Before we continue, Krishna et. al. did not use the same training data set and evaluation data set as the other two papers so that may be the cause of some discrepancies. Ringeval et. al. and the implementation of LBP-TOP done by Sun et. al. have the highest accuracy, around 36%. The main difference of these two papers is that Ringeval et. al. use a gaussian kernel while Sun et. al. use an RBF kernel.

Krishna et. al. discuss that their training data set was produced in a lab under specific conditions while the evaluation data was from movies, where facial images are almost never perfectly aligned images of faces. The SVM was trained to recognize straight-on images of faces with clear backgrounds and since a small amount of the data was like that, it caused a large amount of false negatives. Also, with their system, the accuracy of the emotion recognition depends largely on the accuracy of the extracted facial information and the accuracy of the points that were applied to the image before LBP-TOP was performed. They obviously could not manually apply these points because of the large data sets for both training and evaluation, so the accuracy is off at times, especially when the face is rotated or not viewed straight-on.

Since both Sun et. al. and Ringeval et. al. do a lot more that is beyond the scope of this paper, the discussion from their respective studies is mostly focused on issues in their whole system, while in this paper only the visual portion is discussed.

Future work will probably progress to use more complex machine learning, such as deep neural networks and more advanced networks that would require a lot more processing power than the efficiency of support vector machines currently provide. In the future more papers may take the technique of extracting visual information from the Ringeval et. al. study, using a full-face extraction along with focused extraction of areas that are shown to provide more emotional information, such as the lips, eyes, and eyebrows.

5. REFERENCES

[1] A. Ben-hur and J. Weston. Chapter 13 a userâĂŹs guide to support vector machines.

[2] A. Dhall, R. Goecke, J. Joshi, K. Sikka, and T. Gedeon. Emotion recognition in the wild challenge 2014: Baseline, data and protocol. In *Proceedings of the 16th International Conference on Multimodal Interaction*, ICMI '14, pages 461–466, New York, NY, USA, 2014. ACM.

[3] E. Kim. The kernel trick, 2013.

[4] T. Krishna, A. Rai, S. Bansal, S. Khandelwal, S. Gupta, and D. Goyal. Emotion recognition using facial and audio features. In *Proceedings of the 15th ACM on International Conference on Multimodal Interaction*, ICMI '13, pages 557–564, New York, NY, USA, 2013. ACM.

[5] J. Paivarinta, E. Rahtu, and J. Heikkila. Volume local phase quantization for blur-insensitive dynamic texture classification. In A. Heyden and F. Kahl, editors, *Image Analysis*, volume 6688 of *Lecture Notes in Computer Science*, pages 360–369. Springer Berlin Heidelberg, 2011.

[6] M. Pietikainen. Local binary patterns. *Scholarpedia*, 5(3):9775, 2010. revision 137418.

[7] F. Ringeval, S. Amiriparian, F. Eyben, K. Scherer, and B. Schuller. Emotion recognition in the wild: Incorporating voice and lip activity in multimodal decision-level fusion. In *Proceedings of the 16th International Conference on Multimodal Interaction*, ICMI '14, pages 473–480, New York, NY, USA, 2014. ACM.

[8] B. Sun, L. Li, T. Zuo, Y. Chen, G. Zhou, and X. Wu. Combining multimodal features with hierarchical classifier fusion for emotion recognition in the wild. In *Proceedings of the 16th International Conference on Multimodal Interaction*, ICMI '14, pages 481–486, New York, NY, USA, 2014. ACM.

[9] K. Teknomo. Introduction to svm tutorial, 2007.

[10] G. Zhao and M. Pietikainen. Dynamic texture recognition using local binary patterns with an application to facial expressions. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 29(6):915–928, June 2007.