LBCNNSVM: Local Binary Convolutional Neural Network Support Vector Machine for Facial Expression Recognition

Rushil Patel
School of Computer Science
and Applied Mathematics
University of Witwatersrand
Johannesburg, South Africa
1679175@students.wits.ac.za

Ritesh Ajoodha
School of Computer Science
and Applied Mathematics
University of Witwatersrand
Johannesburg, South Africa
ritesh.ajoodha@wits.ac.za

***Abstract*—Our problem domain is to classify one of the seven human expressions (anger, disgust, fear, happy, neutral, sad, surprise) given an image. Convolution Neural Networks (CNN) are very good at extracting robust information from images and generally perform well on their own. To accelerate the extraction process and point a CNN in the direction of the problem domain we can utilize Local Binary Patterns (LBP). Given our problem LBPs are ideal since they capture texture information. For classifications tasks in CNNs and other networks a softmax activation function is used for prediction and minimizing a cross entropy loss. We show that having a SVM layer that uses a Radial Basis Function (RBF) kernel instead of softmax gives us a slight advantage**.

I. INTRODUCTION

Speech, writing, body language, gestures/actions, and facial expressions all aid in human communication. The most popular method of communication with a computer is through text and speech. A new level of human-computer connection is added when a computer is equipped with the capacity to identify facial expressions. One category of expression recognition is verbal, while the other is non-verbal. Verbal is expressing yourself orally and verbally communicating.
In non-verbal communication, the focus is on the body language, and for this paper, the face in particular.

We see in [1,5] that feature extraction is crucial for success. Since CNNs have gained popularity in the computer vision space for extracting robust features and achieving high accuracies, the focus has shifted from using hand crafted features. But even though hand crafted features are sometimes considered obsolete, we explore some of its advantages in this project by using LBPs of an image.

The focus of this research is to combine the two most common machine learning methods used in the image recognition and classification domain, which are CNNs and SVMs and deliver a hybrid model called LBCNNSVM which uses LBP for feature refining and emulates a SVM in the last layer of a CNN.

II. BACKGROUND AND RELATED WORK

A. Image Pre-processing

A digital image is constructed with a two-dimensional matrix of values that represent the intensity of light. Image pre-processing is the

process of preparing the images from the dataset so that they are optimal for extracting important information. This process can include removing unnecessary data, known as noise. Noise can occur from simply the camera itself or external factors like the weather. Noise in images inhibits machine learning models to accurately model what is in the image, and this leads to a drop in the performance of the model. Noise can be reduced by blurring the images with different intensities.

Colour images are comprised of three channels, Red, Blue, Green. The colours we see in any image is a variable combination of these channels. It can be imagined that three two-dimensional matrices stacked on each other, for example, green and blue gives us the colour yellow. If we use colour images as our input, we would need to consider all three channels of light, increasing the complexity of the model. Instead, we reduce the number of dimensions to just one by converting the colour format to grey scale as done by [3]. Another process is centring and scaling of each image to achieve consistency in the input. This can be done by simple geometric transformations.

B. Local Binary Pattern (LBP)

Our brain does all this processing very easily and fast. But how can a computer pick out specific regions of the face. Reference [6] looks at two kinds of parameters, real valued and binary valued. Some real valued parameters consisted of the distance of the upper eyelid to the eyebrow or the distance between the two lips to determine mouth width. Some of the binary parameters were if teeth were visible or not or if forehead lines were visible or not.

Local Binary Patterns are most common for capturing texture information. This information can be very useful in the setting of analysing faces since we are interested in the minor details that shape an expression, as the aforementioned real and binary values. It is also seen in [5] and [1] that LBP brings a boost in performance when faces are involved. This is achieved using (1).

$$LBP = \sum_{n=1}^{N} S\,(g_n - g_C) * 2^n \qquad (1)$$

C. Convolutional Neural Network (CNN)

CNNs are a class of neural networks that are commonly used in computer vision. What distinguishes a CNN from a general neural network is the convolution layers, pooling layers and the non-linearities such as ReLU and sigmoid.

A convolution operation slides a filter over an image and computes the dot product of the input's region and the weights of the filter. This operation then produces a 2-dimensional feature map containing responses of the filter of local regions of the input. This feature map is usually passed into a max pooling layer which reduces the size of the input. This is important since it decreases overfitting and the number of parameters in the network that needs to be learned. Finally different activation functions are used like sigmoid or ReLU to introduce non-linearities. The main objective of a CNN is to extract features which can later be used in a fully connected layer.

Finally, the learnt features are passed through a fully connected layer which typically uses a softmax function for classification using (2). In this paper we take a different approach and replace the softmax layer with a SVM for classification, which is further discussed in later sections.

$$\sigma(z)_i = \frac{e^{z_i}}{\sum_{j=1}^{K} e^{z_j}} \qquad (2)$$

D. Support Vector Machine (SVM)

Linear data can be easily separable with a line and SVMs are extremely good at doing this [2]. They can also be extended to non-linear data. This is done with the "kernel trick". A kernel can be used to project the data into n-dimensions. A very popular kernel function used is the Radial

Basis Function (RBF) shown in (3) which makes it easier to introduce a hyperplane between points as illustrated in Fig 1.

$$K(x, x') = e^{(-\frac{(x-x')^2}{2\sigma})} \quad (3)$$

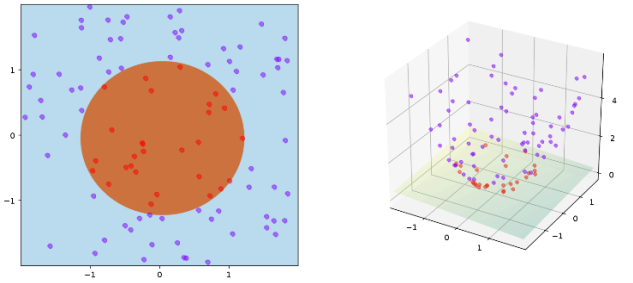

Fig 1: RBF kernel applied to non-linear data

After the data is mapped to n-dimensions, linear SVM can be performed as usual by finding n-dimensional hyperplanes.

Our problem requires a multiclass classification and the simplest way to extend a SVM for multiclass problems is using the one-vs-rest approach. This means breaking the problem into multiple binary class classifications. It must be noted that softmax and multiclass SVMs are almost the same except that softmax maximizes the log-liklehood and SVMs try to find a maximum margin that separates data points of different classes [8].

III. METHODOLGY

A. Research hypothesis

Using a composite of convolutional neural networks and support vector machines using local binary patterns will increase classification accuracy.

B. Dataset

The FER-2013 dataset used, obtained from Kaggle [9] which consists of 35000 grayscale images of size 48x48 pixels. The images are spread across seven classes of expressions shown in Fig 2. In terms of the shape and orientation of the

Fig 2: Training data. Each row consists of expressions: starting from the top row: Angry, Disgust, Fear, Happy, Neutral, Sad, Surprise.

images, they are all standardized, but the pixel values which range from 0 to 255 are standardized between 0 and 1.

C. LBCNNSVM

Our proposed hybrid approach is inspired by [5,8]. Hence, we applied LBP to images that were used for training before feeding them into the network.

The following architecture was implemented for the CNNSVM part of the model with activation function ReLU throughout the whole network:

INPUT: 48 x 48 x 1
CONV: 3 x 3 size 32 filters, stride 1
MAXPOOL: 2 x 2 size, stride 2
DROPOUT: $p = 0.2$
CONV: 3 x 3 size 32 filters, stride 1

MAXPOOL: 2 x 2 size, stride 2
DROPOUT: $p = 0.2$
CONV: 3 x 3 size 32 filters, stride 1
MAXPOOL: 2 x 2 size, stride 2
DROPOUT: $p = 0.2$
FC: 256 hidden neurons
FC: 256 hidden neurons
DROPOUT: $p = 0.3$
FC: 128 hidden neurons
SVM: 7 Output classes, RBF kernel

Note the dropout layers, these are used to prevent overfitting by randomly setting input units to 0 with the frequency of p. Inputs not set to 0 are scaled up by 1/(1 - p) such that the sum over all inputs is unchanged.

IV. Experiments and Results

Implementation was done using Google Tensorflow and can be found at https://github.com/RushilPatel0703/LBCNNSVM_Local-Binary-Convolutional-Neural-Network-Support-Vector-Machine. All experiments were done using Windows 10 operating system, Intel(R) Core(TM) i7-8565U CPU @ 1.80GHz, 12 GB of DDR4 RAM, 500 GB SSD.

Evaluation of our model was done by accuracy as well as comparison with another CNN that used softmax and a CNN that used an SVM but not the LBP images.

TABLE I
PERFORMANCE OF MODELS

Models	**Results**
CNN-Softmax	96.52%
CNN-SVM	94.85%
LBCNNSVM	97.60%

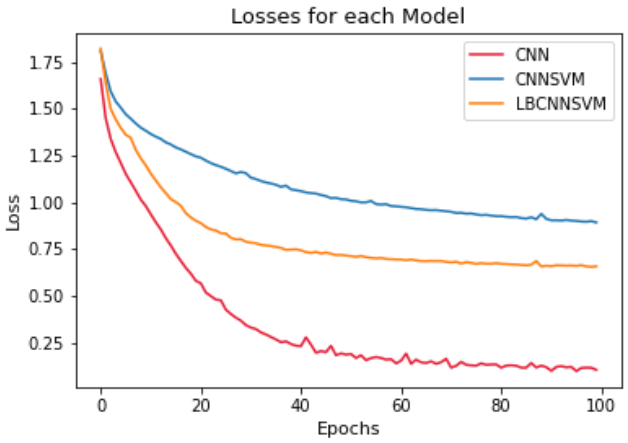

Fig 3: Training loss of CNN, CNNSVM and LBCNNSVM

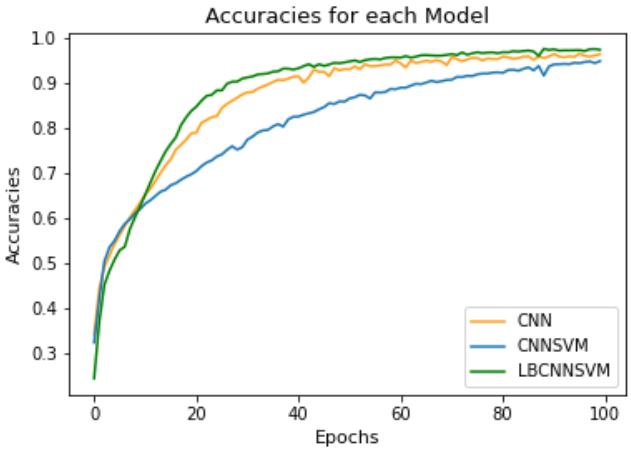

Fig 4: Training accuracy of CNN, CNNSVM and LBCNNSVM

Table I summarizes the performance of our experiment. There is a marginal difference between our approach and a CNN that uses softmax achieving 96.52%.

The goal of using LBP was to accelerate the learning and improve overall performance and this is clearly visible in Fig 4 that using LBPs gives a boost in performance.

We can therefore accept our proposed hypothesis of increasing the prediction accuracy by achieving 97.60%. This shows that using an SVM for classification in deep learning does give an advantage compared to a typical softmax function and is a slight improvement from the results achieved by [8].

Due to time constraints further tuning of LBCNNSVM was not possible as well as training for larger epochs. A general solution has not been achieved with these results as we had constrained our input to just one colour channel rather than all three colour channels of an image. Therefore, our solution can be considered bounded on grayscale images.

V. CONCLUSION

In this report we have shown that when LBP and SVMs are used together we achieve great performance on a standard dataset. Switching from softmax to an SVM in a CNN is not a new idea but we have established its potential. Thus, it is apparent that hand-crafted features are not entirely obsolete as LBCNNSVM achieved top results.

Other ways of learning features and classification in this problem domain can be to use DenseNets. DenseNets are densely connected-convolutional networks, this can be simply thought of multiple CNNS used as a single layer in a network which are all connected that produced exceptional results in [4,7].

For further work, we seek utilizing images with all three colour channels, RGB and other channels like hue, saturation, and value (HSV), to see how much of a gain or loss can be obtained.

ACKNOWLEDGEMENT

An expression of gratitude to Prof Ritesh Ajoodha for his valuable guidance.

REFERENCES

[1] A. I. Maqueda et al, "Human–computer interaction based on visual hand-gesture recognition using volumetric spatiograms of local binary patterns," Computer Vision and Image Understanding, vol. 141, pp. 126-137, 2015.

[2] Cortes and V. Vapnik. 1995. Support-vector Networks. Machine Learning 20.3 (1995), 273–297. https://doi.org/10.1007/BF00994018

[3] H. M. Ariza, H. H. Martínez and L. A. Gaviria Roa, "Recognition system for facial expression by processing images with deep learning neural network," Telkomnika, vol. 17, (6), pp. 2975-2982, 2019

[4] K. Zhang, Y. Guo, X. Wang, J. Yuan and Q. Ding, "Multiple Feature Reweight DenseNet for Image Classification," in IEEE Access, vol. 7, pp. 9872-9880, 2019, doi: 10.1109/ACCESS.2018.2890127.

[5] S. Kumawat, M. Verma and S. Raman, "LBVCNN: Local Binary Volume Convolutional Neural Network for Facial Expression Recognition from Image Sequences," 2019.

[6] S. S. Kulkarni, N. P. Reddy and S. I. Hariharan, "Facial expression (mood) recognition from facial images using committee neural networks," Biomedical Engineering Online, vol. 8, (1), pp. 16-16, 2009

[7] X. Li et al, "Classification of breast cancer histopathological images using interleaved DenseNet with SENe (IDSNet)," PloS One, vol. 15, (5), pp. e0232127-e0232127, 2020.

[8] Y. Tang, "Deep Learning using Linear Support Vector Machines," 2013.

[9] “Challenges in representation learning: Facial expression recognition challenge,” Kaggle. [Online]. Available: https://www.kaggle.com/competitions/challenges-in-representation-learning-facial-expression-recognition-challenge/data.