IN3060/INM460 Computer Vision Coursework report

- **Student name, ID and cohort:** Elnara Mammadova (210026788) - PG
- **Google Drive folder:** https://drive.google.com/drive/folders/1907A-h5wYyQN4DaTpmTSRhFW82ofNK_a?usp=sharing

Data

The dataset contains 12271 training and 3068 test samples of cropped face images (100x100) with labels based on the expressed emotions (7 classes: surprise, fear, disgust, happiness, sadness, anger, neutral). Initial analysis of the dataset showed class imbalance problem, where the distribution of classes was biased (fig. 1). A Synthetic Minority Oversampling Technique (SMOTE) was applied on the training set alone, where images for the minority classes were randomly duplicated. Even though the returned duplicate images were distorted (the image shape had to be flattened before applying augmentation) this technique showed much better accuracies compared to random oversampling technique. The purpose of applying SMOTE was to combat any possible overfitting in our chosen models by adding in more data. Validation set was created from training set to evaluate model performance and perform hyper-parameter tuning. For Support Vector Machine (SVM) k-fold cross-validation was performed where the training set was split into k smaller sets and model evaluated k consecutive times with different splits each time. For Convolutional Neural Networks (CNN), random train and validation subsets were generated (80/20 ratio) using stratify parameter in order to maintain the class distributions amongst both sets. Final distribution of the dataset: train-26723, validation-6681, test-3068.

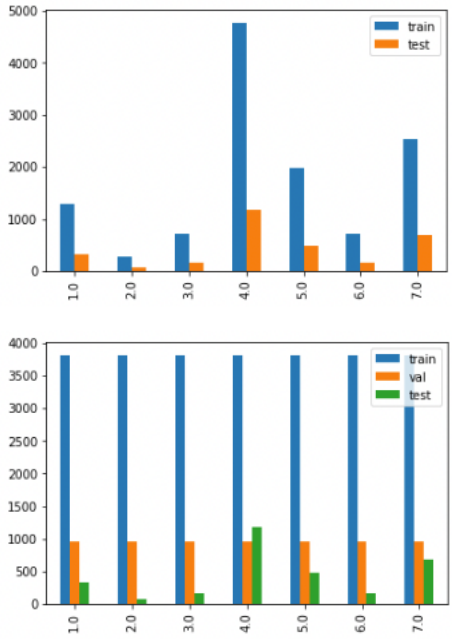

Figure 1. *Data Distribution before and after SMOTE. (bottom figure also includes the stratified train-validation split for CNN model).*

Implemented methods

This project chose three models for critical analysis and evaluation:

1. SIFT descriptors with SVM classifier
2. HOG descriptors with SVM classifier
3. Convolutional Neural Network

SIFT and HOG are the choice of traditional feature descriptors used in this study, which are algorithms that generate numerical vectors to aid in feature differentiation. SIFT is an algorithm that transforms an image into a large collection of 128-dimensional local feature vectors that summarises a 16x16 window with 4x4 bins per window and 8 orientation channels. HOG is very similar to SIFT, where it counts the gradient orientation in localized form, however unlike SIFT, it is purely gradient based, but is great at capturing edges and corners of an image. We use a single model (SVM) in combination with both feature descriptors in order to provide a better comparative analysis between SIFT, HOG and CNN. While both SIFT and HOG have been successful in employing image gradients to describe local structures, the introduction of CNN has made them (amongst other traditional feature descriptors) absolute. CNN is a hierarchical deep learning architecture with multiple trainable convolutional filters which makes it highly adaptive. It can also learn low-level features similar to SIFT and HOG with minimal feature engineering.

SIFT-SVM implementation steps: (1) localising interest points and extracting a list of feature descriptors from each image of the training set; (2) generating Bag of Visual Words (BoVW) vocabulary of descriptors, aka codewords, using k-means clustering; (3) training SVM model using RBF (Radial Basis Function) kernel function and histogram of codewords as inputs; (4) hyper-parameter tuning using 5-fold cross-validated grid search; (5) testing the final model on the hold out test set.

HOG-SVM implementation steps: (1) extraction of HOG feature descriptors using 8 orientation bins and 8x8 window of 1x1 bins with no block normalization – the main purpose being to efficiently be able to differentiate between HOG and SIFT descriptors; (2) train SVM model using RBF kernel function and HOG descriptors as inputs; (3) hyper-parameter tuning using 2-fold cross validated grid search; (4) testing of the final model on the hold out test set.

CNN Implementation steps: (1) transform image data to tensors and apply normalization to the image in the range of [-1, 1] – normalization helps CNN perform better, by getting data within a range and reduce the skewness which helps it learn faster and better; (2) construct CNN model architecture (fig 2.)– the model architecture is similar to LeNet [1] (3) manual hyper-parameter tuning for model parameters (batch size, layer numbers, learning rate, optimizer, criterion, weight-decay/momentum); (4) testing of the final model on the test set.

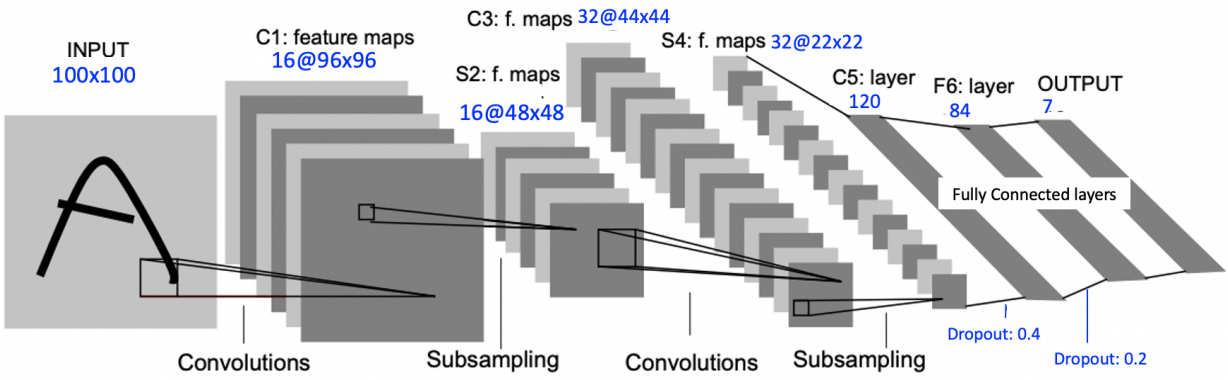

***Figure 2**. CNN architecture adapted from LeNet (LeCun et al. 1998). The descriptions in blue are the modified parameters for the architecture used in CNN FER analysis of this project. (Input size, 100x100 images of 3 rgb channels; Kernel size- 5x5; Conv 1st layer – 16 kernels, Conv 2nd layer – 32 kernels; MaxPooling over 2x2 window; 2 additional dropout layers between second MaxPool layer and first fully connected layer, and first fully connected layer and second fully connected layer in order to reduce overfitting and improve generalization error.)*

Results

Based on the test results, CNN was the best performing out of the three models, with the best accuracy metrics. SVM trained on HOG features performed much better than SVM trained on SIFT features, with accuracy scores closer to CNN.

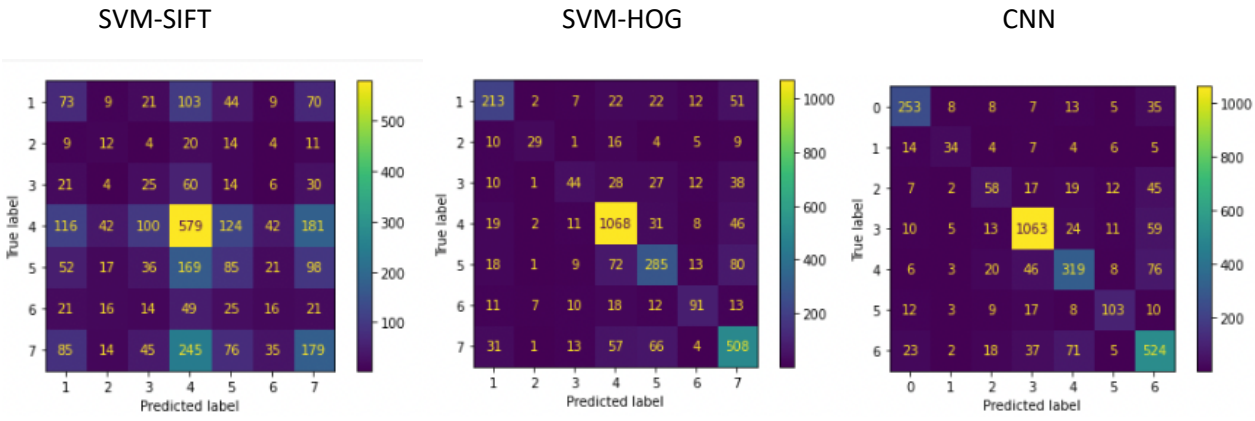

***Figure 3.** Confusion matrix for test results corresponding to three models.*

Table 1. Accuracy metrics for three models

			surprise	*fear*	*disgust*	*happiness*	*sadness*	*anger*	*neutral*	*total*
SVM	*SIFT*	*precision*	0.19	0.11	0.10	0.47	0.22	0.12	0.30	
		recall	0.22	0.16	0.16	0.49	0.18	0.10	0.26	
		f1	0.21	0.13	0.13	0.48	0.20	0.11	0.28	
		accuracy	0.22	0.16	0.16	0.49	0.18	0.10	0.26	0.31
	HOG	*precision*	0.68	0.67	0.46	0.83	0.64	0.63	0.68	
		recall	0.65	0.39	0.28	0.90	0.60	0.56	0.75	
		f1	0.66	0.50	0.35	0.87	0.62	0.59	0.71	
		accuracy	0.65	0.39	0.28	0.90	0.60	0.56	0.75	0.73
CNN		*precision*	0.78	0.60	0.45	0.89	0.70	0.69	0.69	
		recall	0.77	0.46	0.36	0.90	0.67	0.64	0.77	
		f1	0.77	0.52	0.40	0.89	0.68	0.66	0.73	
		accuracy	0.77	0.46	0.36	0.90	0.67	0.64	0.77	0.76

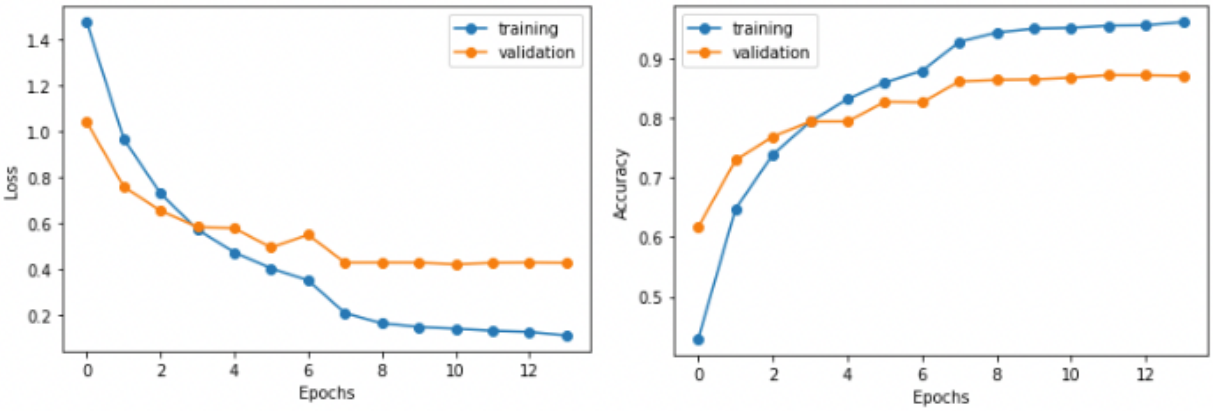

Figure 3. *CNN training results*

Figure 3. *CNN training results on video "in the wild".*

For video "in the wild" project we have chosen a ~4 minute video clip from Mr. Bean movie, and we applied our pre-trained custom CNN model for FER analysis. Since the CNN was trained on dataset with predominantly happy, neutral and disgust emotion classes, most of the video features were identified as such. Even though the data was augmented (using SMOTE) to account for the class imbalance, it still resulted in biased predictions.

Discussion

- As seen on the figures above, HOG-SVM results in better predictions than SIFT-SVM. Since SIFT uses gaussian smoothing to weight the values of 16x16 descriptors, it introduces an additional bias which isn't present in HOG. Hence, SIFT is more suited to identifying objects, as it helps locate the local features (keypoints). HOG on the other hand is better suited in classification tasks, as it extracts global features. In the case of FER analysis, HOG computes edge gradient of the whole image and generates histogram with orientations of each pixel, which is better at capturing facial expressions and differentiate between features.
- The best hyperparameter combinations for SVM obtained through cross validated grid search for SIFT descriptors were C=10 and gamma at scale, whereas for HOG the C=1000 and gamma at scale. Since we were utilising RBF kernel the default value of parameter gamma at scale (1/num_features * X_variance) works well for both descriptors. C on the other hand is the parameter that regularizes the model, and with imbalanced dataset high C values can improve performance.
- CNN had 1883487 total trainable parameters. Initial training results from LeNet structure with 6 and 16 kernels for each Convolutional layers resulted in poorer performance, than the architecture using 16 and 32 kernels. Intuitively the more complex the dataset we would expect the network with more kernels to perform better. The use of two dropout layers for CNN improves the generalization of the networks while also reducing overfitting by preventing units from co-adapting too much [2]. Dropout is more effective where there is limited amount of training data.
- SIFT results can be improved further using interest points extracted using a fixed grid. The problem with the imbalanced class associated with the dataset further limits the classifier's performance for minority classes. A further augmentation could be performed using under sampling for clustering based on Gaussian Mixture Model (GMM) on top of the SMOTE technique.

References

[1] Y. Lecun, L. Bottou, Y. Bengio and P. Haffner, "Gradient-based learning applied to document recognition," in Proceedings of the IEEE, vol. 86, no. 11, pp. 2278-2324, Nov. 1998,

[2] Srivastava, N. et al. (2014) "Dropout: A Simple Way to Prevent Neural Networks from Overfitting," Journal of Machine Learning Research. 15. pp. 1929-1958.