IN3060/INM460 Computer Vision Coursework report

- **Student name, ID and cohort:** Igor Pereira Martins (190053025) - UG
- **Google Drive folder:** https://drive.google.com/drive/folders/1RQ4UZicR8JVDich_YEaRvLQKQvfEqan2?usp=sharing

Data

Provided set

The provided data set consisted of 12271 training images and 3068 testing images, this provided a split of around 20% testing, which is generally considered a good amount for datasets of this size. This also provided enough data to achieve reasonable accuracy even without optimized hyperparameters.
There were, however, several problems with the provided dataset: several images here heavily edited in order to be the right size as well as rotation, this reduced the quality of these images as well as creating a lot of blank space, which prevented the feature descriptors from performing their best, and reduced the overall accuracy of the model; The amount of images in each class varied by a large margin, for example, there were 4,772 images in the happiness class, making 38.8% of images be from this one class, where an equal distribution should have had it closer to 14.2%, this lead to models erroneously predicting a lot of images as happiness.

Personal set

The personal data set consisted of 21 testing images taken from CC search, 3 images for each class, these images were in both colour and black and white as well as representing several age groups, and lighting conditions to test the models with a large variety of images.
A larger dataset would have been better to truly showcase the accuracy of the model, however compiling a dataset of a large enough size was not plausible within the time constraints.

Implemented methods

Model 1: SIFT SVM with BOVW

The first model I implement used a SIFT identifier and a SVM classification model, this was done as a starting point to base my decisions for my other models as this one was found to be the simplest and least accurate model combination.
To train this model, the data was imported using a data loader with a batch size of four.
SIFT was then applied to each image and the descriptors were saved in a list and then run through a BoVW histogram to get the final list that would be run through the SVM with a kernel type of rbf, closely following the example provided by lecture 7.
Applying SIFT took around 90 seconds, however, creating the histogram took 15 minutes, increasing the total time taken to train the model.

Model 2: HOG SVM

The second model used a HOG identifier and a Linear SVC classifier model, a type of SVM, I chose this model so that I could compare HOG to SIFT and test which identifier produces better results.
To train this model, the data was imported using a data loader with a batch size of four, no validation set was used as cross validating SVMs took too much processing time and the results without one were already very high.
HOG was then applied to all images in both the testing sets and training sets, this took 20 minutes and 3 minutes respectively whilst training the actual model was quite quick, making the total time taken quite long despite SVMs typically being quite quick to train.

Model 3: CNN

The final model was a Convolutional Neural Network, this was done to test whether a more complex type of model would provide better results than an SVM.
To train this model, the data was imported using a data loader with a batch size of 4 and the training set was split into training and validation to a ratio of 9:1.
Only 6 filters were applied to the inputted channels, and then 16 were applied to outputs of that layer, this matches the amount used in Tutorial 8, there was an attempt to use 32 filters for the first layer to provide

higher accuracy, however, this increased training time for each epoch from 3 to 90 minutes and provided a minimal increase in accuracy, and trying to use something in between stopped the model from working, so the filters were kept as they were.
SoftMax was not used for the output of the model as it reduced the accuracy by 30% and did not provide any opportunities for more epochs before overfitting.
After 3 epochs the validation loss began to increase so the model was not saved past this epoch to avoid overfitting.

In the wild

The Emotion Recognition function allowed all three of these models to be run on a set of testing images taken from the Wild.
The images themselves were uncropped and of varying sizes, so the first thing done to them before using the classifier to predict their labels was to use the CV2 Haarcascade face detector to find the faces in the images, crop the images and then resize them to 100 by 100 to keep them like the images in the provided dataset.
Several functions were then made, based on the existing code used to train and test the models before implementing Emotion Recognition and are used to further process the image depending on the model the user choses to use.

Results

Model 1: SIFT SVM

The SVM took around 5 minutes to train and applying SIFT to the images took around 10 minutes, however, the model itself performed very poorly, Surprise and Happiness were its best performing classes, and even these had a rather low accuracy, it is likely that increasing max iterations would provide better results, but 100 were already used to train the current model, therefore it is unknown just how many iterations would be needed for passable accuracy to be achieved.

Emotion	Accuracy	Precision
Anger	16.97%	3%
Disgust	5.92%	4%
Fear	1.19%	2%
Happiness	29.35%	42%
Neutral	3.72%	23%
Sadness	13.03%	18%
Surprise	23.07%	10%
Overall accuracy of the network on the test images: 17.9%		

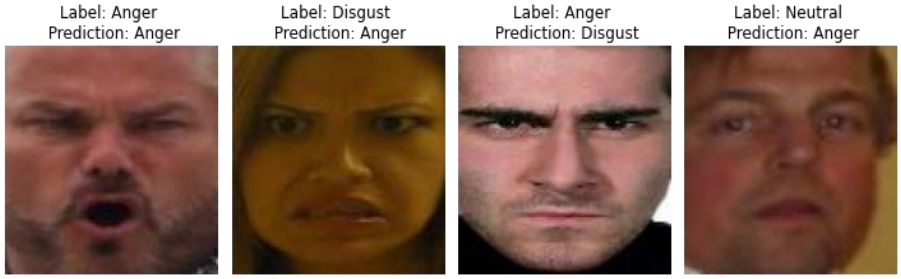

Figure: Qualitative representation of Accuracy of SVM with SIFT on provided data set

Model 2: HOG SVM

The SVM itself only took around 5 minutes to train, however, processing the data itself took a rather long time compared to SIFT, taking 20 minutes, which would likely take even longer if HOG were tweaked to provide more detailed descriptors, which would be one of the few ways to increase accuracy further. Class accuracy was mostly consistent apart from Happiness and Disgust, the former being much higher, likely due to the surplus of data, and the former being much lower.

Emotion	Accuracy	Precision
Anger	58.19%	67%
Disgust	23.39%	30%
Fear	40.63%	59%
Happiness	86.18%	82%
Neutral	61.62%	61%
Sadness	61.34%	57%
Surprise	66.77%	73%
Overall accuracy of the network on the test images: 68.9%		

Figure: Qualitative representation of Accuracy of SVC with HOG on provided data set

Model 3: CNN

The CNN took around 15 minutes to train and provided a very high overall accuracy for the short amount of time spent training, class accuracy fluctuated, however, with Happiness and Surprise being the classes with the highest accuracy whilst Disgust and Fear having the lowest, this likely comes again from the difference in how many examples it had to train on and it is likely that it would produce better results with a more balanced training set.

Emotion	Accuracy	Precision
Anger	56.17%	65.0%
Disgust	28.12%	43.27%
Fear	24.32%	81.82%
Happiness	89.45%	82.62%
Neutral	59.56%	67.61%
Sadness	53.77%	56.48%
Surprise	85.71%	60.65%
Overall accuracy of the network on the test images: 70.34%		

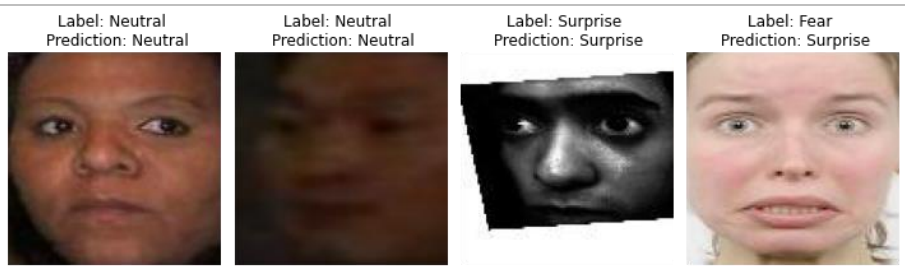

Figure: Qualitative representation of Accuracy of CNN on provided data set

Personal Data Set

As the CNN had that highest accuracy overall it was applied to the personal dataset to test how good the model is in the wild. As shown the accuracy is not nearly as good as the provided dataset, I believe this to be because the data was not processed as much as it could have been, where for the provided dataset the data was rotated and cropped based on distance from the eyes, the personal dataset only used a facial recognition algorithm which sometimes cut off key features of the images.

Emotion	Accuracy	Precision
Anger	33.33%	50.0%
Disgust	0.0%	0.0%
Fear	0.0%	0.0%
Happiness	33.33%	50.0%
Neutral	0.0%	0.0%
Sadness	66.67%	100.0%
Surprise	66.67%	16.67%
Overall accuracy of the network on the test images: 23.81%		

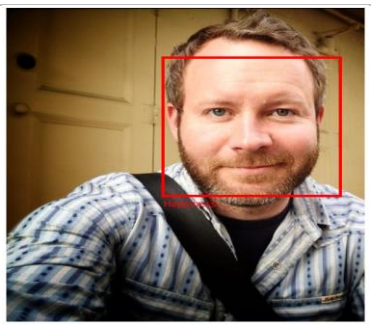

Figure: Bounding box with label

When testing each model on the dataset it appears that using the HOG with SVC model provides better results on the personal dataset than the CNN, however, as the Emotion Recognition function is not optimized for testing there is only qualitative data to go off, but based on the 4 images below, three of which are identical, one can assume that this is true.

Figure: Qualitative representation of Accuracy of CNN on personal data set

Figure: Qualitative representation of Accuracy of SVC with HOG on personal data set

Discussion

Speed

When considering training time, model 1 was the fastest to both train and test, applying SIFT to all images took much less time than applying HOG, allowing for the model to be tweaked much more often and presenting an opportunity to increase the iteration to attempt to increase accuracy, however, even after 100 iterations accuracy did not increase.

Model 3 was the second fastest model, this was likely due to the fact that not identifier was used on the images, cutting out a lot of the training time as the model could begin training straight away with no need to process data, potentially training time could have been increased by adding more layers to the network, which might increase accuracy, however, the increase in accuracy was negligible compared to how much training time increased, making this approach unviable without better hardware.

The slowest model to train and test was Model 2, this was due to HOG taking as long as it did to apply to the data before the model could be run, decreasing the number of cells and pixels per cell could increase speed but would greatly affect the accuracy of the model, better hardware would likely reduce the processing time.

Accuracy

The most accurate model changed between the provided dataset and the personal data set, model 3 was the most accurate by a small margin on the provided set, whilst model 2 was seemingly more accurate on the personal set, I believe this is because the images in the personal set are not processed in the same way as the images in the provided set, on which the two models were trained, possibly showing that the CNN is more rigid when it comes to test data, and if the images in the personal set were processed appropriately then it would perform better, on the other hand, since model 2 was trained on image descriptors, and the personal data had HOG applied before being given to the model, I believe that the model had an easier time finding the common features for each class.

Model 1 performed the worst in terms of accuracy, I believe this is because SIFT works much better with differentiating different objects, whereas the key points for a lot of emotions are very similar and therefore the identifier was unable to find which common key points represented each class.

Precision

In terms of precision both model 2 and model 3 had very similar results, which matches up with their accuracy, however, model 3 had better precision when it came to classes that had lower accuracy, showing that model 3 had a more robust understanding of what to look for in its lower accuracy classes.

Model 1 had a low precision overall, but happiness was it's highest, I believe this is because of how skewed the data was towards the happiness class, with it having several times more data than some class, it is likely that balancing the training set would provide better results for both this model and the other two.

Overall

In conclusion, the best performing model would be model 2, when you don't consider the one-off training time, which is several times that of model 1 and 3, the accuracy of the model is almost as high as model 3, and it also performs better at classifying images in the wild.

Despite this, I believe that model 3 is a strong contender, and that, had the images in the personal dataset been processed in the exact same way as the provided set, it would have been the best performing model, alongside this, the model itself took very little time to train, and further tweaking may increase its accuracy beyond that of model 2.

Model 1 performed poorly in all aspects apart from speed, leading me to believe that SIFT is not a good identifier for this particular task.