APPLIED MACHINE LEARNING SYSTEM ELEC0132 19/20 REPORT

SN:16077278

ABSTRACT

The report presents solutions for four facial feature classification tasks. It introduces the general topic of facial recognition and facial detection. A convolutional neural network model is presented as a solution to gender classification of 178x218 facial images with 93%(±1%) test accuracy, its hyper parameter tuning, and network architecture detailed. The same neural network is retrained to classify the same images based on whether or not a smile is detected in which an 84%(±3%) accuracy is reached. Support vector machine models are presented for cartoon face shape and eye colour classification achieving 100% and 83% (±1%) accuracy respectively.[1] (GitHub, google drive)

1 INTRODUCTION

Face detection is the field within computer vision which is concerned with the detection of faces in images or videos. The goal of Face detection in machine learning is to identify that there is a face present in the image, but it is not necessarily concerned with identifying additional features of the face. The 2017 Journal Article from the International conference of inventive Systems and Control [1] presents the most commonly used face detection algorithms.

Face Detection is the first step within Facial recognition. Facial recognition is the field concerned with classifying facial images. It enables the identification of features such as gender, age, skin colour, identification of a person and so forth.

Facial recognition is an active research topic within the field of machine learning. It has made its way into multiple commercial applications and it is present in people's everyday lives. Examples of this would be Apple's FaceID technology, Windows Hello face unlock, Automatic passport control gates at airports, Snapchat filters and so forth.

This Report presents possible solutions developed by the author to four tasks which fall under the broad topic of facial recognition. The four tasks are:

- Gender Classification of human face images
- Smile Detection in human face images
- Face Shape Detection of cartoon faces
- Eye Colour Detection in cartoon faces

The tasks where solved using two main machine learning models: Support Vector Machines (SVM) and Convolutional neural networks (CNN).

SVMs classify inputs by trying to find the boundary between the given classes such that with the largest possible margin between the boundary and any given class is the largest possible. SVMs can fit linear and non-linear decision boundaries between classes.

Neural Networks (NN) consist of interconnected nodes ("neurons"), organised into layers, which act as small mathematical computational units changing the information passing through the layer. These are initialised at random, once data propagates through the network once, the weights of the nodes are adjusted to reduce the difference between the output of the network and the expected output.

CNNs are neural networks, that have one or multiple convolutional layers as their input layer. A convolutional layer differs from a regular NN layer because it filters the input for features and gradually learns which features to extract, and it is very effective at extracting features from a images.

CNNs were used in solving the Gender Classification task and the Smile Detection task. The Smile Detection was attempted both with CNNs and SVMs. While for solving the tasks involving cartoon faces, only SVMs where used, after initial test with CNNs were unsuccessful.

The report is structured into INTRODUCTION which touches upon the broad topic of Facial recognition, introduces the tasks solved presented by the report, describes the high level concept behind the main machine learning methods used within the report.

The next section is a Literature survey which is an overview of the potential approaches to each task. It introduces both classical machine learning practices and the latest solutions within each field.

[1] Link to the project https://github.com/kazmer97/Machine-learning-assigment

Description of Models section contains the detailed description of each model's architecture and explains the rationale behind choosing each model for the given task.

It continues with the Implementation which describes how each mentioned model was implemented, by explaining the operation of the key modules of the code.

This is followed by the presentation of Experimental results and Analysis where the performance of the models are presented and analysed.

The report ends with a Conclusion section in which the key results of the experiment and analyses are summarised along with suggestions for future improvement.

2 LITERATURE SURVEY

Face detection has been present in the machine vision research since the late 1990s, and by now it is available for usage in multiple libraries. The detection algorithms are usually SVM or Tree based, using general geometrical features of the human face [2][3] [4].

Gender classification has been achieved quite a while a ago. Prior to 2012, the state of the art models were SVM based. An example for SVM based gender classification is detailed in a paper published in 2000 by B. Moghaddam in which SVM outperformed human test subject in classifying the gender of 21x12 human face images. Past 2012, the development of convolutional neural network models to image recognition problems has exploded due to advances in computing and neural network theory. CNN have been shown to outperform traditional methods in accuracy and training time needed[5][6].

Octavio Arriaga proposes a real time convolutional neural network for emotion and gender classification in its 2017 paper [7], with promising results. The emotion recognition capability of CNNs seems to indicate that program should be able to recognise smiles in pictures as well.

The jury is still out whether or CNNs or SVMs do a better job at the isolated task of smile detection, but it very much depends on the input data and benchmarks. Example of this is an article from 2009 on an SVM smile detection system [8] which points out the bias in face detection algorithms between white and black faces (around 7% accuracy drop for black faces) in common libraries such as OpenCv. The drop in face detection accuracy reduces the smile detection accuracy of their system as well.

Cartoon face recognition is a lot less researched topic than human face recognition. Due to the lack of large datasets from before and the failure of traditional methods to classify them. There is no agreed upon best practice for cartoon face detection or classification, so far as cartoon faces can differ significantly between each other. A 2018 paper written by Suarav Jha explores the several deep learning methods and demonstrates that Inception neural network combined with an SVM establishes state of the art results in gender classification of cartoon faces [9].

Kohei Takayama proposed a different face detection method in 2012 based extracting the features from images based on colour information, separating the faces based on a skin colour HSV range and conducting symmetry tests on the extracted mask to determine whether it is face [10]. The method shows promising results, but its flaw is that colour schemes can vary widely in cartoons as such one trained model would only be highly effective with faces from the same style of cartoon.

Based on the reviewed literature, human face gender and smile detection is a very actively researched topic with high accuracy results in controlled settings and increasingly good results in uncontrolled settings as well. This is due to the tasks strong commercial appeal in areas like security, consumer electronics, entertainment, photography so forth.

Cartoon face classification is a less explored topic, since successful algorithms in this area have been developed much later than for face recognition and the commercial applications of the results are much more limited.

3 DESCRIPTION OF MODELS

When choosing the models for each task the objectives were:

- To achieve the highest possible accuracy[2] and the lowest possible loss[3] on the test data in case of a CNN model
- To achieve the highest possible accuracy and recall-score[4] on the test data in case of an SVM.

The constraints faced while choosing a model consisted of:

- Model complexity limitation due to hardware constraints (laptop, no dedicated GPU)
- Extent of programming knowledge of author
- Time constraint of the project

3.1 Task A1: Gender Classification

The model used for the task is a CNN which have been shown to perform well on image classification [1], [11]. Taking into account the computational limitations of a laptop, to achieve reasonable training time (less than 30 seconds/epoch) to allow for debugging and experimentation, the images used as an input to the CNN where downsized to 50x50 from 178x218. The colour was kept as it can be a relevant feature for distinguishing male and female faces. The images were not cropped in order to only include the face, since that would have lead to leaving out the persons' hair, which can be a useful feature when detecting their gender.

The model architecture was inspired by general CNN architectures like AlexNet[12] and LeNet[13]. The ordering of the layers is similar, but the number of neurons is scaled

[2] The percentage of correct classifications made by the model.
[3] Loss is the summation of the error made by the network during operation
[4] Recall score is the percentage of correct classification for one class. Further detail [20]

down due to the smaller size of input images and because it only needs to learn to classes as the task is a binary classification problem (face in the picture is male or female).

The model was picked out of 27 model architecture variations which were trained for 10 epochs, ones exhibiting the lowest validation loss was chosen for hyper-parameter tuning.

The input of the model is a three-channel colour picture resized to 50x50 dimensions.

The A1 model contains:

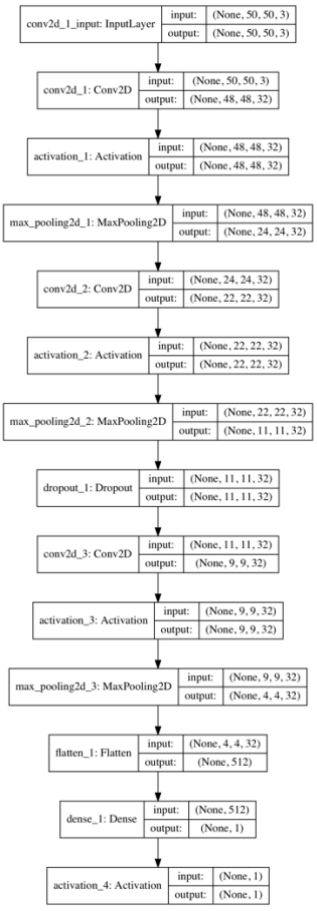


Figure 3.1 Model for Task A1

1. 32 two-dimensional convolutional filters of size 3x3x3, moving with 1 stride. Followed by a rectified linear operator (ReLU) and a two-dimensional maxpooling layer taking the maximim value of 2x2 regions.
2. The 32x24x24 output of first layer is processed by the second convolutional layer, which is followed by the same activation function and maxpooling layer as the first.
3. The 32x11x11 output of the second layer is processed by the third one which has the same activation function as before and maxpooling layer.
4. The output of the third layer is flattened and passed into a fully connected layer.
5. The fully connected layer has one output. And uses a Sigmoid activation function.

See Figure 3.1. for the Keras model summary.

The model is compiled with an Adam optimiser [14][15] which was shown to perform well with CNNs and converges towards potential solutions quicker, reducing the computational cost.

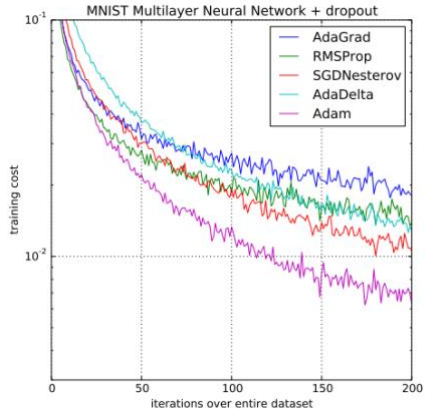


Figure 3.2 Training cost of Adam optimiser according to Adam paper [14]

The loss function was specified as “binary_crossentropy”, which along with the Sigmoid activation function is a strong combination for binary classification problems.

3.2 Task A2: Smile Detection

After testing multiple network architectures, the CNN from A1 was retrained for task A2 as well. The justification behind it was that from the variety of models the accessible computer could run in a reasonable time the CNN architecture from A1 achieved the lowest validation loss. Using the same CNN for both tasks enables a comparison on how well the proposed CNN architecture performs when it has to extract different features from the same dataset.

A competing idea was to use an SVM model which would receive the facial coordinates extracted from the images using the Dlib [16] library, but its implementation was not successful enough, thus the CNN was chosen as it enabled reaching higher accuracies.

3.3 Task B1: Cartoon Face Shape Detection

The model chosen for the face shape detection was an SVM classifier. The model was chosen based on the fact that the input data is very uniform, meaning all the faces are relatively the same size, centred and have a white background. SVMs are good at classifying special points and the face shapes could be classified by learning that where the white pixels switch to colour is where the faces start.

The kernel, C and gamma values were chosen based on a grid search performed on a subset of the training data.

The best performing combination was found to be:

- Kernel: Linear
- C: 0.01
- Gamma: 0.1

3.4 Task B2: Eye Colour Detection

For the cartoon character eye colour classification task was done using an SVM classifier as well as the colours have distinct pixel values, the faces can easily be cropped to zoom in on the eyes so the classifier can learn that the important features are the eyes within the picture.

The kernel, C and gamma values were chosen based on a grid search performed on a subset of the training data.

The best performing combination was found to be:

- Kernel: Linear
- C: 0.1
- Gamma: 0.1

4 IMPLEMENTATION

The code for the project is organised into 4 files. See below the main classes and methods used within the files.

1. Data_Processing.py
    a. Data_A
        i. preprocessing_CNN_A1

ii. preprocessing_CNN_A2
iii. face_detection_dlib

b. Data_B
i. process_B1_svm
ii. process_B2_svm

2. ModelA.py
a. Model_A1
i. find_best_CNN
ii. gender_CNN_refine
iii. gender_CNN_final
iv. test_gender_CNN_final

b. Model_A2
i. find_best_CNN
ii. smile_CNN_refine
iii. smile_CNN_final
iv. test_smile_CNN_final

3. ModelB.py
a. Model_B1
i. find_face_shape_svm
ii. train_face_shape_svm
iii. test_face_shape_svm

b. Model_B2
i. find_eye_svm
ii. train_eye_svm
iii. test_eye_svm

4. main.py.

Everything in Data_Processing is used for the preprocessing of the data for each task. ModelA and ModelB contain the final models and the code used for the finetuning of the parameters. main.py is where the functions from the other modules are called.

4.1 Task A1: Gender Classification

4.1.1 The Dataset

The dataset for the task contains 5000 colour images in jpeg format of size 178x218. The dataset is shuffled by default, meaning both genders occur in random order between the image named 0 to 5000. Further investigation of the dataset reveals that the final 1000 images contain slightly more men than women. The dataset is also balanced containing exactly 2500 images for each gender.

4.1.2 Train, Validation, Test Split

When splitting the data into training, validation and test sets the 60%-20%-20% rule was applied. The last 1000 images are separated to be used as a test set. The first 4000 images then are split by the validation split function present in the model.fit() function within the Keras Python library. The validation split then separates the specified percentage of the passed dataset from the datasets end, into validation set, in this case 25% of it, which account for 20% of the whole data. The described process splits the initial dataset into the above-mentioned ratio.

4.1.3 Main Libraries and Modules used in the code

The OpenCV [17] library is used within the Data_Processing.Data_A.preprocessing_CNN_A1 to load the images to memory and resize them. Within the method the NumPy [18] library is used to create a multidimensional array that will form the inputs to the CNN. The method preforms the splitting of the data into training and test set.

The numpy arrays created by the method containing all the image data has the shape of 5000x50x50x3 (number of images / height of image / width of image / 3 colour channels). The elements within the array copied straight from the images would be between 0-255 (pixel values), but the array is divided by 255 so all elements are between 0 and 1 in the returned arrays.

ModelA.Model_A1.find_best_CNN iterates through 27 possible generic CNN architectures for the task that can be used with 50x50 colour images as input. The key library used within ModelA is Keras [19].

The building blocks of the CNNs are:

- 2D Convolutional layer 3x3 block size, stride 1, padding set to valid, with ReLu activation function, followed by 2D MaxPooling layer with 2x2 block and stride 1
- Fully connected layer

During the iterations all combinations are tried varying the following parameters:

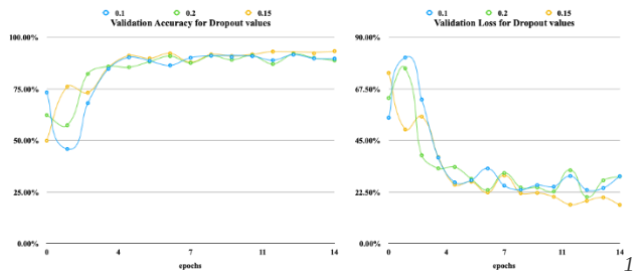
- layer size: 32, 64, 128
- number of convolutional layers: 1, 2, 3
- number of Dense (fully connected) layers: 0, 1, 2

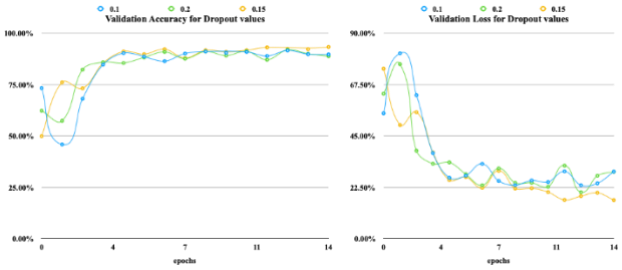
The generated models all run for 10 epochs with a batch size of 32 and a validation split on the received training data of 25%. The training is interrupted if the validation loss hasn't improved in more than two epochs.

The function logs the trainings to Tensorboard, from where the most promising ones can be chosen for further development.

The model chosen to be further developed in for A1 was the one described in section 3.1. Which was recreated in ModelA.Model_A1.gender_CNN_refine method. Within this it is rerun for further parameter tuning, increasing convolutional layer filter sizes, introducing padding, rerunning the model with multiple values for learning rate. Introducing a dropout layer before the third convolutional layer and retraining the model with multiple dropout values.

Based on the results presented in 1





0.001 was chosen as the desirable learning rate for the model.

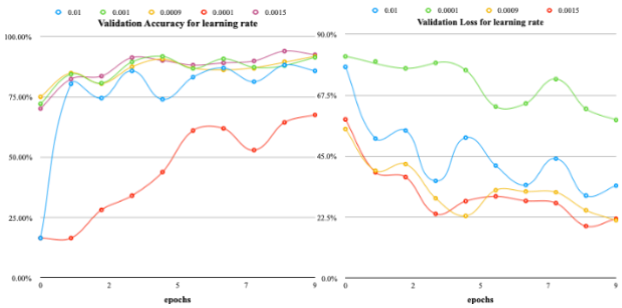


Figure 4.1 Learning rate's effect on Model A1

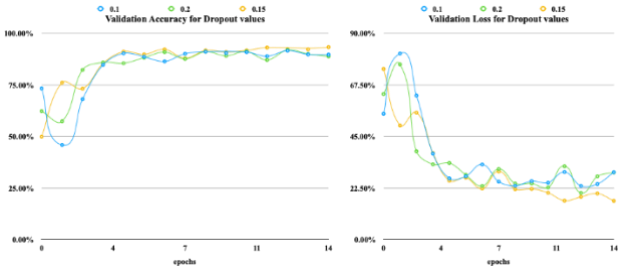


Figure 4.2 Dropout value effect on Model A1

The value of the dropout layers value was chosen to be 0.15 based on the results in Figure 4.2

ModelA.Model_A1.gender_CNN_final contains the final CNN with the decided parameters, which can be trained. The training is allowed to run longer (25 epochs). The progress of the training is saved to Tensorboard, and there is an early stopping criterion implemented which stops the training if the validation loss has not improved in more than 5 epochs. The method also saved the best iteration of the model during training based on the validation loss.

ModelA.Model_A1.test_gender_CNN_final loads the saved model from the disk and evaluates it over the test set separated in the beginning (Test set 1) and the one provided later (Test set 2).

4.2 Task A2: Smile Detection

4.2.1 The Dataset

The dataset consists of the same images that where described in section 4.1.1. The difference is that now they are labelled as 'smile' or 'no smile'. The dataset is balanced 50% of pictures containing a smile while the otherscontaining no smile. The dataset is shuffled in regard to the 'smile' label as well.

4.2.2 Train, Validation, Test Split

The dataset is split in the exact same manner as described in section 4.1.2. With regard to the 'smile' label.

4.2.3 Main Libraries and Modules used in the code

The Data_Processing.Data_A.preprocessing_CNN_A2 method performs the same task as the similarly named method for A1 described above, with one addition. Within the method the dlib [16] library is used for face detection in images, where a face is detected on the imported image, the image is cropped to the face before it is resized. Dlib uses a Histogram of Oriented Gradients (HOG) face detection method. The built-in frontal face detector function returns four coordinates that form a bounding box for all the faces detected within an image by the function.

The rest of the modules serve the exact same purpose as described in section 4.1.3 with the difference that within the name, gender has been replaced with smile, and the model parameters are different.

From the output of the find_best_CNN method it is visible that the larger CNNs overfit the data quickly due to the small scale of the input images but they do so slower than in Task A1 and none of the CNNs breach the 90% accuracy barrier without further parameter tuning.

Based on task A1 and retesting the learning rate parameter in the smile_CNN_refine method, the chosen learning rate is 0.001, which is the default of the Adam optimiser.

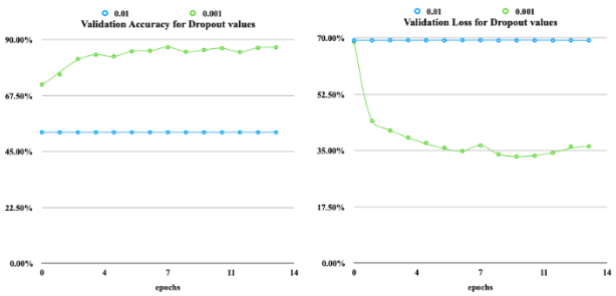


Figure 4.3 Learning rate effects on Model A2

4.3 Task B1: Face Shape Detection

4.3.1 The Dataset

The dataset is made up of 10,000 500x500 colour images in Portable Network Graphics (PNG) format. The dataset is balanced and shuffled. There are exactly 2000 examples of all four faces shapes.

4.3.2 Train, Validation, Test Split

The train and validation split are performed slightly differently compared to task A. The classifier is trained on 60% of the dataset and validated on 20% of it as during the parameter tuning of the SVM. And an additional 20% of the dataset was retained for testing the final model.

4.3.3 Main Libraries and Modules used in the code

The main libraries used in the methods for Task B1 are Scikit-learn [20] for SVM training and data splitting. OpenCv [17] for image reading and NumPy [18] for creating input arrays.

Data_Processing.Data_B.process_B1_svm reads in the images in colour. It resizes each image to 100x100, then the 3

dimensional NumPy array (100x100x3) containing the pixel data for the images is reshaped into a one-dimensional array (30000) that is suitable as input for an SVM. The image data is then added to an array that will contain all the one-dimensional image data. The method then splits the array containing the image data into training and test sets.

ModelB.Model_B1.find_face_shape_svm takes 2000 images and runs an extensive grid search to find the best parameters for the SVM. Those parameters are then set in the ModelB.Model_B1.train_face_shape_svm, which trains the model on 6000 images, tests it on 2000 and saves the final model.

ModelB.Model_B1.test_face_shape_svm loads the saved model and tests it on 2000 images that were separated as test set (Test set 1) and not used at all during parameter search and also tests it on the test set that was provided later during the course of the assignment (Test set 2).

Table 4.1 Summary of Model parameters at the end of development

	Training Accuracy	Training Loss	Validation Accuracy	Validation Loss / Error
A1	0.97	0.0828	0.932	0.1641
A2	0.8875	0.2718	0.8425	0.3869
B1	-	-	1	0
B2	-	-	0.82	0.18

4.4 Task B2: Eye Colour Detection

4.4.1 The Dataset

The dataset is the same set of images as the ones described in section 4.3.1, only the labels are different this time, as they refer to the eye colour. The dataset is balanced and shuffled, however, it cannot be considerd clean. Some of the images are labelled as a certain eye colour, but the eye is not visible in the picture as the person is wearing black sunglasses.

4.4.2 Train, Validation, Test Split

The train, validation, test split is performed the same way as described in section 4.3.2.

4.4.3 Main Libraries and Modules used in the code

The modules follow the same logic using the same libraries as the ones described in for Task B1, with the exception that during the data processing the images are cropped around the eyes and after the cropping they are not resized anymore.

The classification report on the validation set

	precision	recall	f1-score	support
0		0.74	0.84	387
1		0.86	0.83	414
2		0.82	0.78	384
3		0.81	0.82	406
4		0.85	0.81	409
accuracy			0.82	2000
macro avg	0.82	0.82	0.82	2000
Weighted avg.	0.82	0.82	0.82	2000

5 EXPERIMENTAL RESULTS AND ANALYSIS

Table 5.1 Summary of model performances on test sets for Each task

Model	Test set	Test Accuracy	Test Loss
A1	Test set 1	91.6%	0.238
	Test set 2	94.4%	0.162
A2	Test set 1	83.4%	0.3788
	Test set 2	87.4%	0.2981
B1	Test set 1	100%	0
	Test set 2	100%	0
B2	Test set 1	84%	16%
	Test set 2	82%	18%

Table 5.2 Task B2 Classification report Test set 1

	precision	recall	f1-score	support
0	0.76	0.86	0.81	365
1	0.87	0.83	0.85	438
2	0.84	0.81	0.82	385
3	0.84	0.82	0.85	413
4	0.88	0.84	0.86	413
accuracy			0.84	2000
macro avg	0.84	0.84	0.84	2000
Weighted avg.	0.84	0.84	0.84	2000

Table 5.3 Task B2 Classification report for Test set 2

	precision	recall	f1-score	support
0	0.77	0.83	0.80	506
1	0.84	0.82	0.83	483
2	0.82	0.81	0.82	525
3	0.81	0.82	0.81	514

4	0.86	0.83	0.85	472
accuracy			0.84	2500
macro avg	0.82	0.82	0.82	2500
Weighted avg.	0.82	0.82	0.82	2500

From the results, it is apparent that the CNN used in A1 and A2 performs better when it needs to differentiate between gender, while it cannot extract the features necessary to detect smile.

The lower performance in A2 is likely due to the low resolution of the images passed as input, as the features in an image indicate a smile are much more subtle and are confined to fewer pixels than the features indicating one's gender.

However, using low resolution input images allows the model to generalise on high level features better. During future work to improve the performance of these models, the trained model A1 and A2 could be reused during transfer learning to extract high level features as part of a larger neural network capable of extracting more features from the images.

The performance of A2 could be further improved by adding the facial landmarks returned using a Dlib library to the input data.

The SVM in B1 performed very well and there doesn't seem to be much space for further improvement as it classified all the faces correctly for each class 100% of the time for both test sets. That is why no classification table was included separately for it as it would show two tables containing only 100% values.

The performance of the SVM model in B2 could be improved by cleaning the input data. In order to achieve this, an algorithm has to be written, which can detect the people wearing black sunglasses and remove these images from the dataset.

6 CONCLUSION

Task A1 and B1 were solved with a fairly high accuracy, using models generally applied to these tasks 93% and 100% accuracy respectively. The solutions to task A2 and B2 could use further improvement. The A2 model might not be the best choice, but it shows that a CNN structure that does well on the gender classification is not able to extract all the necessary features to detect a smile with the same accuracy. The B2 model is likely to reach over 90% accuracy by improving the data cleaning process. Unfortunately, the error of the black sunglasses being labelled as different eye colours was detected too late in the course of the assignment to correct it by the time of the deadline. Lastly the main area that falls under future work is implementing a grid search with Scikit-learn and Keras to formalise the hyperparameter tuning of neural network models

7 LIST OF ACRONYMS

1. SVM – Support Vector Machine
2. CNN – Convolutional Neural Network
3. NN – Neural Network
4. ReLU – Rectified Linear Operator
5. HOG – Histogram of Oriented Gradients

8 REFERENCES

[1] A. Srivastava, S. Mane, A. Shah, N. Shrivastava, and B. Thakare, "A survey of face detection algorithms," *Proc. Int. Conf. Inven. Syst. Control. ICISC 2017*, pp. 15–18, 2017.

[2] C. Zhang and Z. Zhang, "A Survey of Recent Advances in Face Detection," 2010.

[3] G. Guo, S. Z. Li, and K. Chan, "Face recognition by support vector machines," *Proc. - 4th IEEE Int. Conf. Autom. Face Gesture Recognition, FG 2000*, no. February 1970, pp. 196–201, 2000.

[4] Q. Chen, H. Wu, and M. Yachida, "Face detection by fuzzy pattern matching," *IEEE Int. Conf. Comput. Vis.*, pp. 591–596, 1995.

[5] G. Antipov, S. A. Berrani, and J. L. Dugelay, "Minimalistic CNN-based ensemble model for gender prediction from face images," *Pattern Recognit. Lett.*, vol. 70, pp. 59–65, 2016.

[6] J. Lemley, S. Abdul-Wahid, D. Banik, and R. Andonie, "Comparison of recent machine learning techniques for gender recognition from facial images," *CEUR Workshop Proc.*, vol. 1584, pp. 97–102, 2016.

[7] O. Arriaga, M. Valdenegro-Toro, and P. G. Plöger, "Real-time convolutional neural networks for emotion and gender classification," pp. 221–226, 2017.

[8] J. Whitehill, G. Littlewort, I. Fasel, M. Bartlett, and J. Movellan, "Toward practical smile detection," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 31, no. 11, pp. 2106–2111, 2009.

[9] S. Jha, N. Agarwal, and S. Agarwal, "Bringing Cartoons to Life: Towards Improved Cartoon Face Detection and Recognition Systems," no. i, 2018.

[10] K. Takayama, H. Johan, T. Nishita, and Takayama et al., "Face detection and face recognition of cartoon characters using feature extraction," *Image, Electron. Vis. Comput. Work.*, p. 48, 2012.

[11] G. Levi and T. Hassner, "Age and Gender Classification using Convolutional Neural Networks," 2015.

[12] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet Classification with Deep Convolutional Neural Networks Alex," 2012.

[13] Y. LECUN, L. ´EON BOTTOU, Y. BENGIO, and P. HAFFNER, "GradientBased Learning Applied to Document Recognition," *Proc. IEEE*, vol. 86, no. 11, 1998.

[14] D. P. Kingma and J. L. Ba, "Adam: A method for

stochastic optimization," *3rd Int. Conf. Learn. Represent. ICLR 2015 - Conf. Track Proc.*, pp. 1–15, 2015.

[15] S. Ruder, "An overview of gradient descent optimization algorithms," pp. 1–14, 2016.

[16] D. E. King, "Dlib-ml: A Machine Learning Toolkit," *J. Mach. Learn. Res.*, vol. 10, pp. 1755–1758, 2009.

[17] G. Bradski, "The OpenCV Library," *Dr. Dobb's J. Sotfware tools*, 2000.

[18] Oliphat and E. Travis, *A guide to NumPy*. Trelgol Publishing USA, 2006.

[19] F. Chollet, "Keras," *https://keras.io*, 2015. [Online]. Available: https://keras.io.

[20] G. Varoquaux, L. Buitinck, G. Louppe, O. Grisel, F. Pedregosa, and A. Mueller, "Scikit-learn," *GetMobile Mob. Comput. Commun.*, vol. 19, no. 1, pp. 29–33, 2015.