Supervised Learning Methods for Facial Recognition and Classification

Student Number 15050994
University College London
https://github.com/UCL15050994/AMLSassignment.git

Abstract

This report presents and compares multiple approaches to solve a number of binary and multi-class face classification facial recognition and classification problems from a noisy 5000–image dataset. Logistic regression and support vector machines (SVMs) were used in order to classify the images after successful outlier removal and their performances were compared. It was found that SVMs are preferable to logistic regression in virtually all of the classification tasks, classifying the images with accuracies ranging from 82% to 99%.

1. Introduction

1.1. Problem Statement

Facial recognition has become an increasingly important and burgeoning area of research over the past few decades and is one of the most crucial applications of machine learning and deep learning. The main aim of this project is to carry out a number of classification tasks on a dataset comprising 5000 labeled images, primarily those of human and avatar faces. The images are preprocessed and drawn from two larger image datasets: "CelebFaces Attributes Dataset (CelebA)", a dataset containing images of celebrity faces [37], and "Cartoon Set", a dataset containing images of cartoon faces [17]. The dataset also contains noisy images that are images of nature and/or blank backgrounds as well as mislabeled images of faces.

The size of the overall dataset is 266.2 megabytes. The images are all of the portable networks graphics (PNG) format, and are all of size 256×256 pixels. The color models used vary from image to image, with some images being RGB images with three channels and others being grayscale images with one channel. The labels for each image are given in a comma-separated values (CSV) file, in which each column corresponds to a different label. The labels are given in an integer format.

An important task that precedes the main tasks of this project, namely the (supervised) classification tasks, is to filter out the aforementioned outliers in the dataset. The approach used to carry out this task is outlined in sections 1.2 and 2.3. Subsequently, the images are to be classified according to the labels in the CSV file. This equates to five classification problems: four binary classification problems and one multi-class classification problem. The classification tasks are as follows:

1. *Emotion recognition*: smile (+1) or no smile (-1).
2. *Age identification*: young (+1) or old (-1).
3. *Glasses detection*: with (+1) or without (-1).
4. *Human detection*: real (+1) or cartoon (-1).
5. *Hair color recognition*: bald (0), blond (1), ginger (2), brown (3), black (4), or gray (5).

1.2. Data Preprocessing

This project involves more than one stage of data preprocessing. The first stage of data preprocessing is the removal of the noisy images/outliers in the dataset. The second stage of data preprocessing involves transforming the image data in different ways in order to extract the features that are to be fed to the classifiers used for training, testing, and validation.

Noise removal was carried out with the help of a frontal face detector from the dlib library (refer to section 3.1) which involves the use of a Histogram of Oriented Gradients (HOG) feature descriptor in conjunction with a linear classifier, a sliding window detection scheme, and an image pyramid. The frontal face detector detector detects and maps 68 facial landmarks in the form of (x, y) coordinates for each image that contains a face. Figure 1 depicts the locations at which the 68 facial landmarks are mapped. As can be seen in figure 1, the landmarks are mapped along the eye, eyebrow, jaw, nose, and mouth areas.

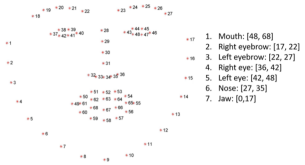

Figure 1. The location of the 68 landmarks [3].

The frontal face detector was used in order to determine whether or not a face was present in each image in the dataset. Images for which no facial landmarks were

detected were labeled as outliers and discarded. In conjunction with the OpenCV library (refer to section 3.1), this process was carried out twice in order to investigate the effect of grayscaling the images prior to filtering outliers. It was found that when grayscaling was applied to the images using the imread function from OpenCV (effectively converting the RGB and RGBA images with three and four channels respectively to grayscaled images with one channel), 585 outliers were detected. When the same process was carried out without grayscaling, 544 outliers were detected. Assuming that a necessary (but not necessarily sufficient) condition for an image to be an outlier is to have all of its associated labels equal to -1, the number of outliers cannot exceed 435. By inspection, it was found that all of the images that were not discarded in both cases were images of faces. Thus, it was inferred that noise removal with RGB images (removing 544 outliers) is more accurate that noise removal with grayscaled images (removing 585 outliers). Therefore, noise removal with RGB images was used to produce the new dataset without outliers on which the classification tasks (and further preprocessing) were subsequently performed. The second stage of preprocessing, namely transforming the image data in order to extract features that can be fed into the classifiers, was carried out in more than one way. One way was to use a modified function from an external module in conjunction with the frontal face detector from the dlib library in order to extract the 68 facial landmarks for each image and store them in an array of arrays. The data was then reshaped into a one-dimensional array as appropriate before training, testing, and validating the classifiers. Another way was to downsample the images from 256×256 pixels to 128×128 pixels and read the raw RGB values for each image using the OpenCV library. Downsampling was carried out since it is less computationally expensive and reduces training times, while preserving enough color information in each photograph. Additionally, for the hair color classifier in particular, the images were cropped such that only the top half of the each image was fed into the classifier. Since images for which the hair color label is -1 are mislabeled, they were eventually discarded while training the final implementation of the hair color classifier. As with the case of the landmarks, the RGB data was reshaped into a one-dimensional array as appropriate before testing.

2. Proposed Algorithms

2.1. Support Vector Machine (SVM)

One proposed algorithm for both the binary and multi-class classification tasks is the use of support vector machines (SVMs). SVMs are supervised learning models that combine the use of a kernel trick with a modified loss function and can be used for classification problems as well as regression problems [29, 22]. Unlike a logistic regression classifier, an SVM outputs a class identity but does not output probabilities [22]. SVMs are highly robust and have been shown to be useful in a wide variety of applications and classification problems [16, 21]. SVMs can be used for both binary classification and multi-class classification, making them good candidates for all of the classification tasks outlined in section 1.1, including hair color classification. Multi-class SVMs can be implemented using "one-versus-all" schemes in which the N–class classification problem is effectively reduced to N binary classification problems, or using "one-versus-one" schemes in which a binary classifier is used to discriminate between each pair of classes without taking into account the other classes, effectively reducing the N–class classification problem to $\frac{N(N-1)}{2}$ binary classification problems [16, 10, 15]. The latter was used despite its $O(N^2)$ complexity in the number of classes N since "one-versus-all" multi-class SVMs have the disadvantage of having their performance performance be reduced as a result of unbalanced training sets [16], and, as shown by Gidudu *et al*. [16], "one-versus-all" multi-class SVMs exhibit a higher propensity of yielding unclassified and mixed pixels when applied to image classification. One important advantage of SVMs is versatility; both linear and nonlinear kernels can be used for the decision function [10, 33]. In this project, both linear and polynomial (order 3) kernels were investigated. Linear kernels are suitable for linearly separable data, whereas polynomial kernels are suitable for linearly non-separable data [18, 1]. Compared to logistic regression, SVMs don't penalize examples where the correct decision is made with a sufficient confidence level and are thus more generalizable and have a lower tendency to overfit data [33, 34]. Additionally, SVMs give sparse solutions when using the kernel trick, and can be more scalable [34]. The SVM hyperparameters that were investigated are discussed extensively in section 3.2.

2.2. Logistic Regression

Another proposed algorithm for both the binary and multi-class classification tasks is logistic regression. Logistic regression is a statistical model that models binary variables using the logistic function (or the sigmoid function) [29, 22], and, unlike an SVM, outputs probabilities and not class identities [22]. An advantage of logistic regression is its efficiency in terms of both time and memory [30, 33]. For a typical Limited-memory Broyden–Fletcher–Goldfarb–Shanno (L-BFGS) solver, logistic regression typically has $O(n)$ complexity, which is comparable to that of a linear SVM, but significantly more efficient that nonlinear SVMs for which the complexity typically lies between $O(n^2)$ and $O(n^3)$ [13], which makes it a good candidate for the classification tasks outlined in section 1.1. In the multi-class case, logistic regression is

generalized to softmax regression [29]. As is the case with SVMs, softmax regression can be implemented using "one-versus-all" schemes or "one-versus-one" schemes. The latter minimizes the cross-entropy multinomial loss fit across the entire probability distribution [9], and was used in this project. The logistic regression hyperparameters that were investigated are discussed extensively in section 3.2.

2.3. Comparing Different Feature Extractions

As disucssed in section 1.2, two methods were primarily used in order to extract the features from the data, namely extracting the facial landmarks and raw pixel RGB values. All of the SVM classifiers and logistic regression classifiers used for the binary tasks were trained using both forms of preprocessed data, except the multi-class classifiers, which were only trained with RGB values from resized and/or cropped images. The first method exploits the HOG feature descriptor, which has been shown to be highly accurate for image registration [14], in order to extract the facial landmarks from the images. This was used to train both the logistic regression and SVM classifiers for all the binary classification tasks. The reason the facial landmarks were thought to be particularly useful in the case of the binary classification tasks was that the landmarks are mapped to 68 points on the face that include facial features that are relevant to and useful for training each of the binary classification tasks. For example, figure 1 shows that the landmarks are mapped to the mouth region of the face. Training the smile classifier with the landmarks is therefore be appropriate. For the human classifier, the landmarks might be useful due to the large variance in the structure of the face between a real human face and an avatar. Likewise, for the glasses classifier, the landmarks mapped to the eye region of the face could be appropriate for training the glasses classifier since different landmarks are mapped depending on whether or not there are eyeglasses in the image. Finally, the landmarks could be appropriate for age classification due to the overall difference in distances between facial features and bone structures between young and old people. The second method, which primarily involves extracting the raw RGB values from all the images after resizing, was used to train both the logistic regression and SVM classifiers for the multi-class classification tasks as well as the binary tasks, albeit more suited to the former than the latter. The purpose of training the binary classifiers using raw RGB values was to compare their performance and complexity with the binary classifiers trained using the facial landmarks. In the case the hair color classifier, the additional step of cropping the images such that only the RGB values for the top half of each image were used for training the hair color classifier was carried out in order to eliminate as much as possible RGB values that are not relevant to hair color; since that hair typically takes up a larger portion of the top half of each image rather than the bottom half. This reduces the number of features and hence reduces the likelihood of bias and overfitting, which is discussed in section 3.3. An example of this is shown in figure 2. Additionally, training the age classifier with the raw RGB values seemed appropriate given the correlation between being old and having gray hair.

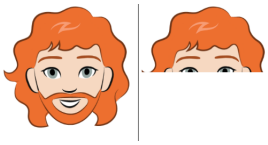

Figure 2. An example of a cartoon image before and after cropping [17].

3. Implementation

3.1. External Libraries

All the design solutions implemented for this project were obtained using the Python 3.6 programming language. A number of external libraries were used in order to carry out both the preprocessing task and the classification tasks in this project. The Scikit-learn library [8] was used in order to implement SVM and logistic regression, splitting the data and labels into training and test subsets as appropriate, plotting learning curves for each classifier, optimizing the hyperparameters for each model and carrying out cross-validation using `GridSearchCV`, and automatically obtaining relevant metrics for each classification task. The OpenCV library [6] was used in order to load image files, read RGB values for each image, and preprocess them as necessary (by grayscaling, resizing, or cropping). The dlib library [2] was used for facial landmark detection by using the frontal face detector that works with a Histogram of Oriented Gradients (HOG) feature descriptor, a linear classifier, a sliding window detection scheme, and an image pyramid as outlined in section 1.2, which was for outlier detection as well as extracting 68 facial landmarks for the training images in order to train the classifiers. Additionally, a modified version of the `lab2_landmarks.py` external Python module [3] was used for convenience in order to simultaneously extract the facial landmarks for each image and its corresponding label, using the `extract_features_labels` function. NumPy [5] and Pandas [7] were used in order to store and manipulate data where appropriate, and Matplotlib [4] was used in conjunction with Sci-kit learn in order to generate the learning curves for each model.

3.2. Data Splitting and Hyperparameter Tuning

Before training the SVM and logistic regression classifiers, the data had to be partitioned into training, testing, and validation subsets (along with associated labels). Initially, 80% of the data was taken to be training data, and

20% was taken to be test data. This partition is commonly used in machine learning literature since the training set has to be larger than the test set in order to make sure that it is representative of the data [32, 11]. For validation, the technique of $k-$fold cross validation was used, a technique that involves shuffling the training data and splitting it into k folds, each of which is iteratively used as a test set whilst the remainder of the data is taken to be the training set in order to evaluate an estimate of the score of the model called the cross validation score which is the mean of the accuracy obtained from each fold [19]. In particular, for this project, 3–fold cross validation was used, and was carried out in conjunction with the hyperparameter optimization described below. The number of folds was limited to three in order to reduce the time for hyperparameter optimization.

Grid search is a method of finding the optimal hyperparameters for a given model by exhaustive search through a given set of combinations of hyperparameters. This technique was used in order to optimize the hyperparameters for each classifier, and was done with the help of `GridSearchCV` [12] from the Scikit-learn library (with which 3–fold cross validation was also implemented). The model with the highest cross validation score is then selected. In some cases, further refinements to the models had to be made after examining their learning curves, as outlined in section 3.3.

For the case of SVM classifiers, grid search was used to pick the best combination out of a linear kernel, a third degree polynomial kernel, and two values of the penalty parameter C: 0.1 and 0.01. The parameter search space had to be limited to just four combinations for this project due to the limitations of the system on which the code was executed as well as the large number of features in the case of RGB data, which would have otherwise extended the time taken for hyperparameter optimization and 3–fold cross validation exponentially. Additional hyperparameters of importance were set to their default values, such as the kernel coefficient γ, which by default is set to be the inverse of the number of features. It should be noted, however, that the value of the hyperparameter γ has no bearing on linear SVMs. The degree of the polynomial was also set to its default value, 3, and was not included was part of the grid search search space due to the aforementioned limitations. It was found that for all the classification tasks except for age and hair color classification, and for both facial landmarks as features and raw RGB values as features, a linear kernel was preferable to a third order polynomial kernel. Additionally, the optimal value of C was found to be 0.01 for all of the SVM classifiers, meaning that a lower penalty on the error term was preferred [10].

Grid search was also used with the logistic regression/ softmax regression classifiers. In particular, three values of the regularization hyperparameter C (analogous to the hyperparameter C for SVMs) were looked at: 0.01, 0.1, and 1. The parameter search space was limited for the aforementioned reasons relating to the number of features as well as the limitations of the computer used to train the models. A Limited-memory Broyden–Fletcher–Goldfarb–Shanno (L-BFGS) solver was used throughout since it is appropriate for large datasets and handles multinomial loss [9], which was used for the multi-class classification problem. The default penalty (L^2–norm) was used for penalization. Additionally, it was found that the hyperparameter corresponding to the maximum number of iterations for the solver to converge, `max_iter`, had to be set to 5000 in order for the L-BFGS solver to converge. Finally, for the multi-class classification task, the `multi_class` parameter was set to `'multinomial'` in order to implement a "one-versus-one" rather than "one-versus-all" scheme. The optimal value of C was found to be 0.01 for all of the logistic regression classifiers except for human classification, and for both facial landmarks and raw RGB values as features. For human classification, it was found that setting the hyperparameter C equal to 1 yields better results, meaning that stronger regularization was not required for human classification in the case of logistic regression.

3.3. Learning Curves

Learning curves were plotted for each classifier in order to assess the training convergence and observe whether or not the classifiers are overfitting. Refer to Appendix B for the five learning curves associated with the five final (best possible) classifiers chosen for each task. Figure 7 shows an ideal fit where there is a gradual decrease in the training score and a gradual increase in the cross validation score; i.e., the cross validation score converges to the training score, at a reasonably high value. Similar patterns are observed in figures A.3 and A.4 in Appendix A. The figure is reproduced below for convenience.

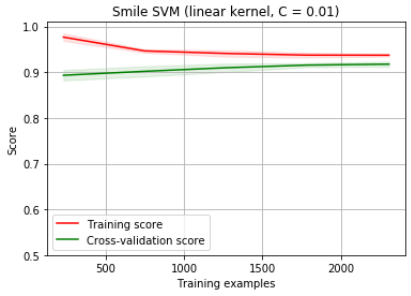

Figure 3. Emotion recognition learning curve with optimized hyperparameters showing training convergence.

On the other hand, figure A.2 shows overfitting: the training score is flat at 100%, while the cross validation score increases very slightly. The large gap between the cross validation score and the training score towards the

right side of the plot is also an indicator for overfitting. Figure A.5 also shows slight overfitting albeit not as pronounced as that in figure A.2. Since overfitting was observed only in classifiers that were trained using raw RGB values, it can be deduced that a cause of the overfitting is using more features than is necessary: raw RGB values for each pixel in the image are more than what is necessary to train a model to perform age classification. For hair color, on the other hand, cropping the images as described in section 2.3 was noted to have reduced the amount of overfitting. This is because raw RGB values for the top half of each image are more relevant to hair color than they are to age.

For all the learning curves, changing the hyperparameter in each classifier corresponding to the tolerance, i.e., the hyperparameter that controls the stopping criterion, had little to no effect, unlike the effect of other hyperparameters. For example, a difference in the training convergence can be denoted in the learning curve for the smile SVM classifier before optimizing the hyperparameters in figure 4 in comparison with that of the hyperparametrized smile classifier in figure 3.

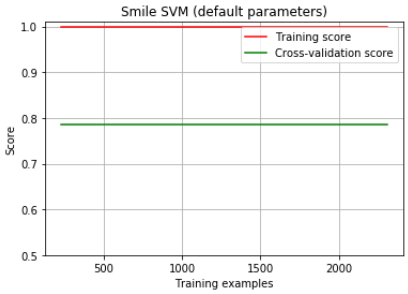

Figure 4. Emotion recognition learning curve before optimizing the hyperparameters showing no training convergence.

4. Experimental Results

In order to compare different classification models, certain metrics are typically used, including precision, recall, the F_1 score, the cross validation score, and the test accuracy score [23]. The final classifiers for each task were eventually chosen primarily based on test accuracy and mean cross validation scores as well as the learning curves, but other metrics are included for all of the 19 hyperparametrized classifiers for the sake of completeness in Appendix C. In addition, a confusion matrix is included in section 4.5 for hair color recognition. The remaining confusion matrices for the binary tasks are omitted for brevity but can be found in Appendix B.

4.1. Emotion Recognition

For emotion recognition, it was found (through the grid search and learning curves) that the best choice of classifier is an SVM classifier with a linear kernel and a penalty parameter $C = 0.01$ trained using the facial landmarks (test score of $\mathbf{91.5\%}$ and cross validation score of $\mathbf{91.8\%}$). This is preferable to using the same SVM classifier trained using the raw RGB values despite having a similar test score since the latter might lead to overfitting and takes a longer time to train due to the larger number of features and significantly higher computational complexity. In addition, the SVM trained using the facial landmarks has a higher cross validation score (91.8%) than that trained using the raw RGB values (89.4%). It also has a slightly larger test score (91.5%) than that with both of the logistic regression implementations (90.6%, 91.0%), and does not show overfitting, as seen in figure A.1 in Appendix A. Testing the classifier on previously unseen and unlabeled test data yielded results that seemed to be in accordance with the test score of 91.5% that was inferred.

4.2. Age Identification

For age identification, it was found (through the grid search primarily) that the best choice of classifier is a logistic regression classifier with an L-BFGS solver and a regularization parameter $C = 0.01$ trained using the raw RGB values (test score of $\mathbf{86.5\%}$ and cross validation score of $\mathbf{85.7\%}$). This was chosen over the SVM classifier trained on raw RGB values despite having a similar a similar test score due to the higher computational complexity of nonlinear SVMs compared to logistic regression with an LBFGS solver. However, testing the classifier on previously unseen and unlabeled test data seemed to yield incorrect results that were not in accordance with the test score of 86.5% that was inferred. This is most likely due to overfitting, as suggested by the learning curve in figure A.2.

4.3. Glasses Detection

Glasses detection was similar to emotion recognition. It was found (through the grid search and the learning curves) that the best choice of classifier is an SVM classifier with a linear kernel and a penalty parameter $C = 0.01$ trained using the facial landmarks (test score of $\mathbf{89.6\%}$ and cross validation score of $\mathbf{88.6\%}$). This was chosen over the SVM classifier trained on raw RGB values despite the latter having a higher test score (test score of 90.1% and cross validation score of 87.3%). Additionally, the model does not show overfitting, as shown in figure A.3. Testing the classifier on previously unseen and unlabeled test data yielded results that seemed to be in accordance with the test score of 89.6% that was inferred.

4.4. Human Detection

For human detection, like glasses detection and emotion recognition, it was found through grid search and learning curves that the best choice of classifier is an SVM classi-

fier with a linear kernel and a penalty parameter $C = 0.01$ trained using the facial landmarks (test score of $\mathbf{99.4\%}$ and cross validation score of $\mathbf{98.5\%}$). The model does not show overfitting, as shown in figure A.4. Testing the classifier on previously unseen and unlabeled test data yielded results that seemed to be in accordance with the test score of 99.4% that was inferred.

4.5. Hair Color Recognition

For hair color recognition, it was initially found (before cropping the images) through grid search that the best choice of classifier is an SVM classifier with a third degree polynomial kernel and a penalty parameter $C = 0.01$ trained using the raw RGB data (test score of 73.0% and cross validation score of 71.9%), performing better than the softmax regression classifier with a test score of 70.6%. The model initially showed clear overfitting however, and testing it on the new test data yielded incorrect results where colors seemed to have been misclassified. The model was then revised in two ways: the images were cropped such that the model was only trained on the top half of each image, and the images labeled with -1 as hair color were filtered out along with their corresponding labels. This new model was found to have a higher test score ($\mathbf{82.1\%}$) and performed well on the previously unseen and unlabeled test data, yielding results that seemed to be in accordance with the new test score of 82.1% that was inferred. This new model and the increase in accuracy corroborates the hypothesis made in sections 1.2 and 2.3; namely that cropping the image enhances SVM training since only the RGB features that are relevant are used for training. Figure 5 shows the confusion matrix for the refined hair color SVM.

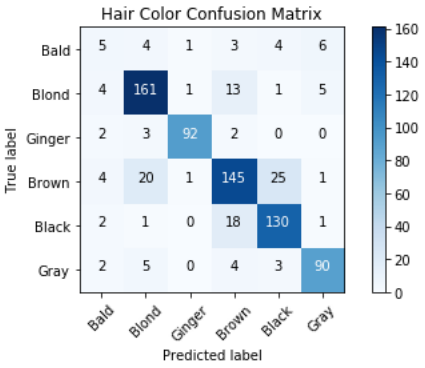

Figure 5. Hair color SVM confusion matrix. The large diagonal terms indicate the extent to which the model classifies a given class correctly whereas the off-diagonal terms indicate misclassification.

5. Related Work

Although the classification algorithms presented in this report belong to the field of classical machine learning, the use of deep learning algorithms and systems such as artifical neural networks for image recognition and in particular facial recognition and classification has been a topic of interest for computer scientists and statisticians alike for much of the 2000s and 2010s. Deep learning methods include convolutional neural networks, recurrent neural networks, and deep belief networks [22]. CNNs are a class of deep neural netowrks (DNNs) that are particularly suited to computer vision and image recognition problems in part owing to their design which is based on a model of the visual cortex of the brain [22]. Additionally, the deep architecture of CNNs allows the extraction of several distinct forms of features at high levels of abstraction [26] making CNNs widely applicable to a number of face recognition applications such as face biometric systems for security [28] and in particular multimodal face biometrics [35] and tackling the issue of time lapses in biometric face recognition systems [24]. A disadvantage of CNNs is that training one requires a large amount of labeled datasets for pre-training that are not readily available [31, 27]. On the other hand, deep belief networks (DBNs) are also widely used for facial expression recognition [20] among other applications, and unlike CNNs, do not required labeled data in a greedy learning process since it is unsupervised [36]. However, training a DBN can be computationally expensive [36], and DBNs generally perform poorly compared to CNNs when tackling computer vision problems [36]. One method that attempts to combine the drawbacks of both CNNs and DBNs is CDBNs [25], which have been shown to have excellent performance on a number of visual recognition tasks as well as the capability to perform hierarchical inference over full-sized images by Lee *et al*.

6. Conclusion

The aims outlined at the start of this project were achieved: noisy images were removed from the dataset by means of a frontal face detector from the dlib library, and subsequent SVM and logistic regression classification of the images using facial landmarks and raw RGB values was carried out successfully for all classification tasks. It was found that except for age identification, SVMs with a linear kernel are superior to logistic regression when trained on facial landmarks from the images, achieving up to 99% accuracy for human and cartoon classification with no significant overfitting. Hair color classification was carried out with a relatively high test accuracy of 82% using an SVM with a polynomial kernel trained on raw pixel RGB values from the resized images in the dataset. Future improvements that could be made to this work include using CNNs in order to carry out hair color classification, and carrying out further preprocessing (such as calculating distances between the facial landmarks used) in order to obtain a more accurate age identification classifier.

References

[1] An idiot's guide to support vector machines (svms), 2008. Stanford University.

[2] The dlib library, 2018. http://dlib.net.

[3] Laboratory exercise 2, 2018. University College London - Department of Electronic and Electrical Engineering.

[4] The matplotlib library, 2018. https://matplotlib.org.

[5] The numpy library, 2018. http://www.numpy.org.

[6] The opencv library, 2018. https://opencv.org.

[7] The pandas library, 2018. https://pandas.pydata.org.

[8] The scikit-learn library, 2018. https://scikit-learn.org/stable/.

[9] sklearn.linearmodel.logisticregression documentation, 2018. https://www.scikit-learn.org/stable/modules/generated/sklearn.linear/linear-model.LogisticRegression.html.

[10] sklearn.svm.svc documentation, 2018. https://scikit-learn.org/stable/modules/generated/sklearn.svm.SVC.html.

[11] Training and test sets: Splitting data, 2018. https://developers.google.com/machine-learning/crash-course/training-and-test-sets/splitting-data.

[12] Tuning the hyper-parameters of an estimator, 2018. https://scikit-learn.org/stable/modules/grid_search.html.

[13] A. Abdiansah and R. Wardoyo. Time complexity analysis of support vector machines (SVM) in LibSVM. *International Journal of Computer Applications*, 128(3):28–34, oct 2015.

[14] E. Abraham, S. Mishra, N. Tripathi, and G. Sukumaran. HOG descriptor based registration (a new image registration technique). In *2013 15th International Conference on Advanced Computing Technologies (ICACT)*. IEEE, sep 2013.

[15] M. Aly. Survey on multiclass classification methods, 2005.

[16] G. Anthony, H. Greg, and M. Tshilidzi. Classification of images using support vector machines, 2007.

[17] M. P. at Google. Cartoon set, 2018. https://google.github.io/cartoonset/.

[18] R. Berwick. An idiot's guide to support vector machines (svms). Massachusetts Institute of Technology.

[19] J. Brownlee. A gentle introduction to k-fold cross-validation, 2018. https://machinelearningmastery.com/k-fold-cross-validation/.

[20] S. I. Ch, N. ng, K. P. Seng, and L. M. Ang. Block-based deep belief networks for face recognition. *International Journal of Biometrics*, 4(2):130, 2012.

[21] O. Chapelle, P. Haffner, and V. Vapnik. Svms for histogram-based image classification, 1999.

[22] I. Goodfellow, Y. Bengio, and A. Courville. *Deep Learning (Adaptive Computation and Machine Learning series)*. The MIT Press, 2016.

[23] J. V. Guttag. *Introduction to Computation and Programming Using Python (MIT Press)*. The MIT Press, 2013.

[24] H. E. Khiyari and H. Wechsler. Face recognition across time lapse using convolutional neural networks. *Journal of Information Security*, 07(03):141–151, 2016.

[25] H. Lee, R. Grosse, R. Ranganath, and A. Y. Ng. Unsupervised learning of hierarchical representations with convolutional deep belief networks. *Communications of the ACM*, 54(10):95, oct 2011.

[26] J. Li, T. Qiu, C. Wen, K. Xie, and F.-Q. Wen. Robust face recognition using the deep c2d-CNN model based on decision-level fusion. *Sensors*, 18(7):2080, jun 2018.

[27] J. Li, T. Qiu, C. Wen, K. Xie, and F.-Q. Wen. Robust face recognition using the deep c2d-CNN model based on decision-level fusion. *Sensors*, 18(7):2080, jun 2018.

[28] S. Malki and L. Spaanenburg. Cbas: A cnn-based biometrics authentication system. In *2010 12th International Workshop on Cellular Nanoscale Networks and their Applications (CNNA 2010)*, pages 1–6, Feb 2010.

[29] K. P. Murphy. *Machine Learning: A Probabilistic Perspective (Adaptive Computation and Machine Learning series)*. The MIT Press, 2012.

[30] A. Navlani. An idiot's guide to support vector machines (svms), 2018. https://www.datacamp.com/community/tutorials/understanding-logistic-regression-pythonl.

[31] M. Oquab, I. Laptev, J. Sivic, M. Oquab, I. Laptev, J. S. Learning, T. Mid-level, M. Oquab, L. B. I. Laptev, and J. Sivic. Learning and transferring mid-level image representations using convolutional neural networks. In *In CVPR*, 2014.

[32] N. Pentreath. *Machine Learning with Spark - Tackle Big Data with Powerful Spark Machine Learning Algorithms*. Packt Publishing - ebooks Account, 2015.

[33] L. Sachan. Logistic regression vs decision trees vs svm: Part ii, 2015. https://www.edvancer.in/logistic-regression-vs-decision-trees-vs-svm-part2/.

[34] K. Swersky. Support vector machines vs. logistic regression. University of Toronto CSC2515 Tutorial.

[35] L. Tiong, S. Tae Kim, and Y. Man Ro. Multimodal face biometrics by using convolutional neural networks. *Journal of Korea Multimedia Society*, 20:170–178, 02 2017.

[36] A. Voulodimos, N. Doulamis, A. Doulamis, and E. Protopapadakis. Deep learning for computer vision: A brief review. *Computational Intelligence and Neuroscience*, 2018:1–13, 2018.

[37] S. Yang, P. Luo, C. C. Loy, and X. Tang. From facial parts responses to face detection: A deep learning approach, 2015.

Appendix A - Learning Curves

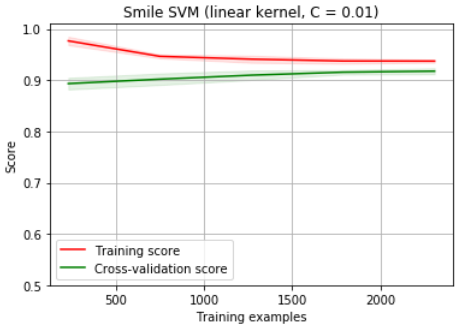

Figure A.1: Learning curve for emotion recognition.

Figure A.2: Learning curve for age identification.

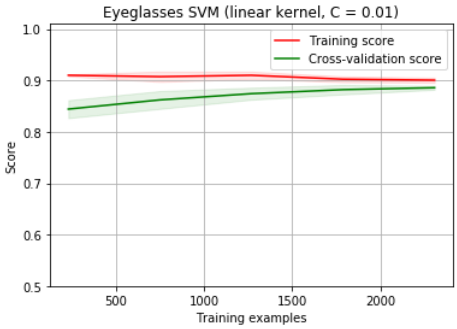

Figure A.3: Learning curve for glasses detection.

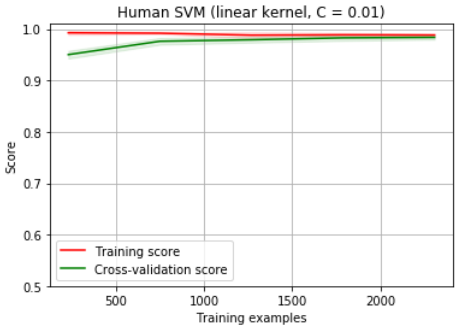

Figure A.4: Learning curve for human detection.

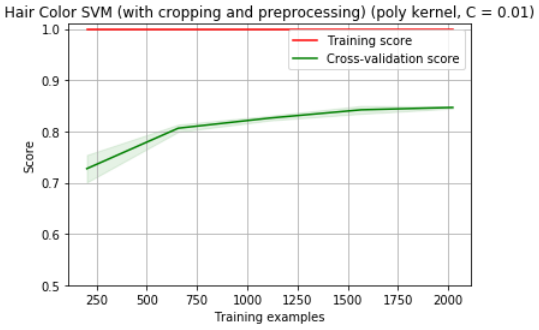

Figure A.5: Learning curve for hair color recognition.

Appendix B - Confusion Matrices

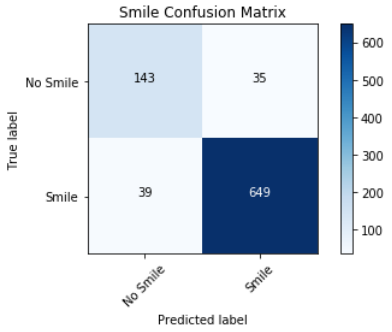

Figure B.1: Confusion matrix for emotion recognition.

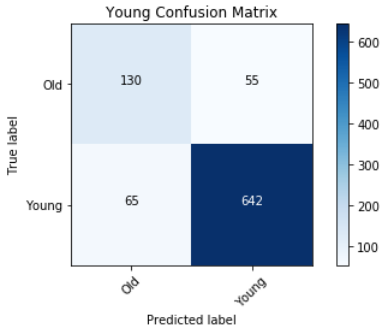

Figure B.2: Confusion matrix for age identification.

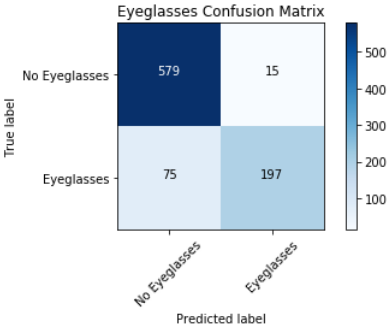

Figure B.3: Confusion matrix for glasses detection.

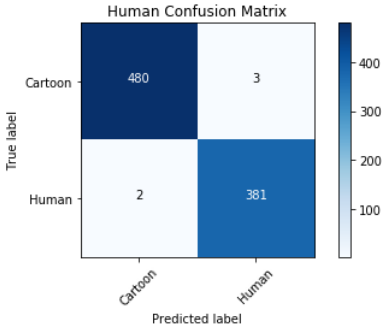

Figure B.4: Confusion matrix for human detection.

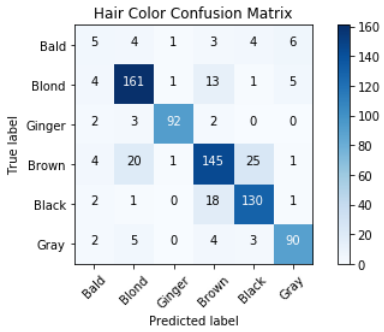

Figure B.5: Confusion matrix for hair color recognition.

Appendix C - Additional Metrics

Classifier	Precision (%)	Recall (%)	F_1 Score (%)	Cross Validation Score (%)	Test Score (%)
Emotion Recognition					
Landmarks SVM	92	91	91	92	91
RGB SVM	90	90	90	87	90
Landmarks Logistic Regression	90	90	89	88	90
RGB Logistic Regression	91	91	90	88	91
Age Identification					
Landmarks SVM	62	79	70	80	79
RGB SVM	86	86	86	85	86
Landmarks Logistic Regression	74	79	73	80	79
RGB Logistic Regression	87	87	87	86	87
Glasses Detection					
Landmarks SVM	90	90	89	89	90
RGB SVM	90	90	90	97	90
Landmarks Logistic Regression	74	79	73	80	79
RGB Logistic Regression	91	91	90	87	91
Human Detection					
Landmarks SVM	99	99	99	99	99
RGB SVM	99	99	99	99	99
Landmarks Logistic Regression	99	99	99	98	99
RGB Logistic Regression	99	99	99	99	99
Hair Color Recognition					
RGB SVM (with cropping)	82	82	82	81	82
RGB SVM (no cropping)	74	73	74	72	73
RGB Softmax Regression	71	71	71	69	71