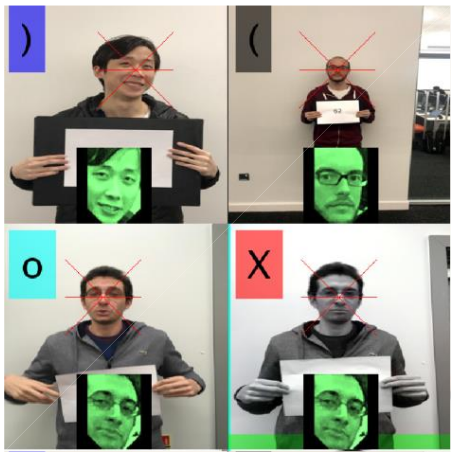
INM 460 COMPUTER VISION: FACE RECOGNITION AND OCR

RICHARD MANN, MSC DATA SCIENCE, CITY UNIVERSITY OF LONDON.



CONTENTS

1 INTRODUCTION

The report provides information on the experimentation, favoured approach and results of a project to develop face image detection and classification function for Computer Vision coursework. The report is structured into sections on Face Detection, Face Recognition, Emotion Recognition and OCR.

Documentation on the two function scripts are provided in sections 2.9 and 3.5.

An overview of the test scripts and test image files also provided with this report is provided in section 4.

All coding for this project was performed in Matlab and use of specific Matlab functions are highlighted in the text below. General Matlab documentation was helpful but the tutorial on 'Face recognition with computer vision' ("Face Recognition - MATLAB & Simulink," n.d.) was most useful in providing a structured approach to face recognition and detection.

2 FACE AND EMOTION RECOGNITION

2.1 FACE DETECTION – OVERVIEW

Face detection was performed primarily using the Viola-Jones algorithm in Matlab's 'CascadeObjectDetector' function. This function was found generally to be reliable, although some optimization was required utilising different classification models ('FrontalFaceCART', 'ProfileFace' etc) and adjusting the threshold for declaring a positive result based on multiple detections around an object ('MergeThreshold').

Two stages of the end-to-end project involved face detection from images, requiring two different approaches:

1. Training data preparation: As discussed further below, building the creation of an image library necessitated face detection from both (a) individual labelled images and (b) unlabelled group images. In this 'off line' process, precision was less important than sensitivity as false positives could be removed manually.
2. On-line face recognition: The face detection for 'live operation' on a test image required a more sophisticated optimization strategy to minimise false positives as well as maintain flexibility to an unknown input image format (size, content etc). A single set of Viola-Jones parameters was not found to be sufficiently accurate therefore a two-stage process was adopted for live operation as follows:
    a. multiple attempts with Viola-Jones using different models and merge thresholds;
    b. inclusion of a 'not-a-face' class in all classifier models, trained on a selection of Viola-Jones false positives found in off line face detection. This ensured that images incorrectly identified in the first stage of face detection could be rejected in the classification process.

2.2 FACE DETECTION – FINAL APPROACH

The final approach adopted for face detection was as summarised below.

2.2.1 Training data preparation

- Most images were detected using the FrontalFaceCART model, with a MergeThreshold of 4 and a detection size of 50 to 600 pixels. All detected faces were resized to 100x100 pixels for standardisation.
- Problematic images were re-processed with ProfileFace and UpperBody detection models.
- For group pictures, the same process was followed but with minimum detection size reduced to 40 pixels.
- For video files, one frame was extracted at 0.5 second intervals (0.25 seconds for students with less available content) and the same face extraction process followed as for JPG images, although very blurred images were rejected.
- False positives, such as those in figure 1, were added to an 'error' category for later classifier development.

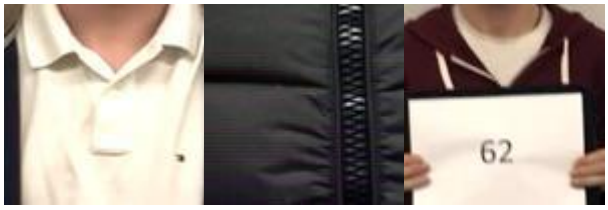


Figure 1: Examples of false positive 'faces' used as '999' category for classifier

2.2.2 On-line face recognition

Experience from the off-line process above informed the development of an automated optimization process for on-line face extraction. Further improvements were made following testing on portraits, group images and images without faces, resulting in the process illustrated in figure 1 below. Stage 1 used an iterative process based on the Viola-Jones classifier and stage 2 used a trained negative class within the face classifiers to reject false positives.

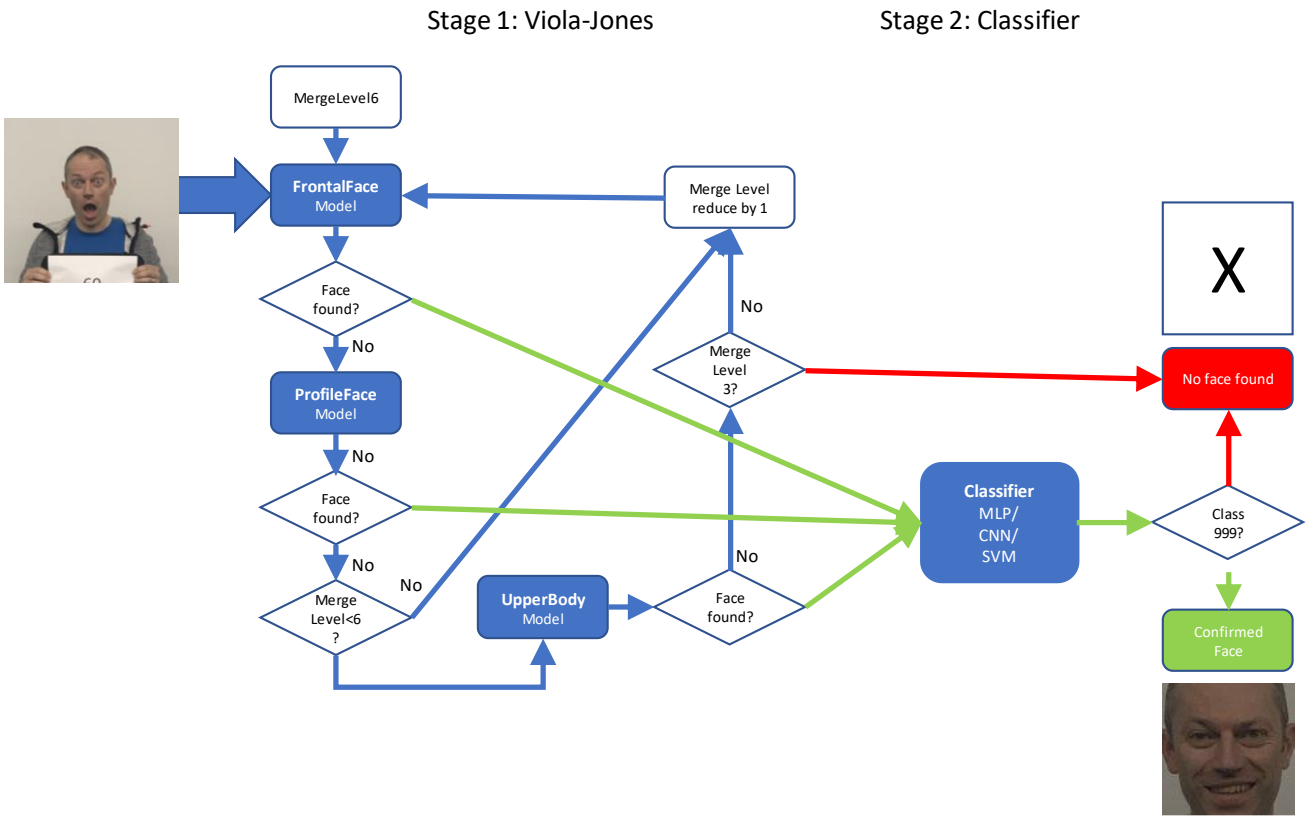


Figure 1: On-line face detection process.

To further help reduce false positives, the following techniques were also adopted:

- Where multiple faces were detected in the process above, iteration stopped as soon as one candidate face from stage 1 successfully passed stage 2.
- UpperBody detections were rejected unless a further EyePairSmall model detection found a positive result.

2.3 Face Recognition – Overview

Experimentation with face recognition optimization evolved through the following stages:

Stage 1: Balancing image data quantity by student

Students #64 to #70 had fewer images because no videos were taken. Additional training images were therefore created with a simple augmentation strategy of resizing images by +/- 10 and 20% and rotating (10^o and 15^o left and right), ensuring a minimum of 20 images per student.

After this stage, the image library consisted of 661 images from faces and 1378 images from video (2039 images total).

Stage 2: Image augmentation and first attempt ("v1") classifiers

Data augmentation, the process of generating samples by transforming training data with the target of improving the accuracy and robustness of classifiers (Fawzi, Samulowitz, Turaga, & Frossard, 2016), was considered necessary as the dataset was relatively small.

Image augmentation was therefore performed on each image as follows:

- Blurring by reducing the size of the image and then scaling back-up to 100x100 pixels
- Adjusting image brightness to mimic changes in lighting conditions

The augmented dataset was split 80% for training and 20% for validation. Classifiers were then built and optimized on the augmented dataset and a simple error visualisation tool was developed to facilitate function optimization as shown in figure 2.

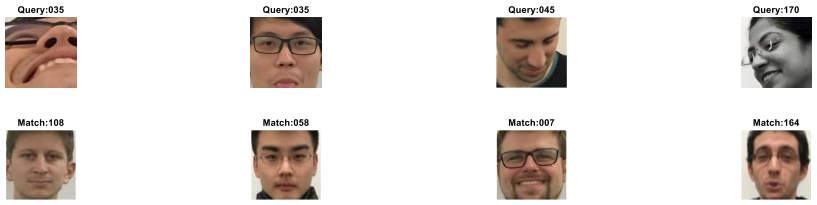


Figure 2: visualisation of validation errors with v1 classifier using SVM and HOG features.

Optimization was performed across the following features:

Model	Parameters optimized	Matlab function used
MLP	1 or 2 layers Layer size 50-200	trainNetwork
CNN	2 or 3 convolutional layers 1 fully connected layer: 50-200 Filter size 4x4 to 6x6 Number of filters 32 to 64	trainNetwork
CNX **(CNN with transfer learning from Alex Net)**	None. AlexNet layers (:end-3) plus one fully connected layer	trainNetwork
SVM	Linear, polynomial 2-4, Gaussian Box Constant and Kernel Size auto-optimised	Fitcecoc

Feature extraction	Parameters optimized	Matlab function used
HOG features	Cell size 6x6 to 10x10 nBins 4 to 9	extractHOGFeatures
Bag of features	Grid step 6x6 to 10x10 Bag of words 500-1000 Blockwidth not varied (default = [23 64 128 256])	bagOfFeatures

Table 1: feature optimization

High levels of validation accuracy (98%+) were achieved for all classifiers. However, classification accuracy was found to be poor (<50%) on unseen group images as shown in figure 3.

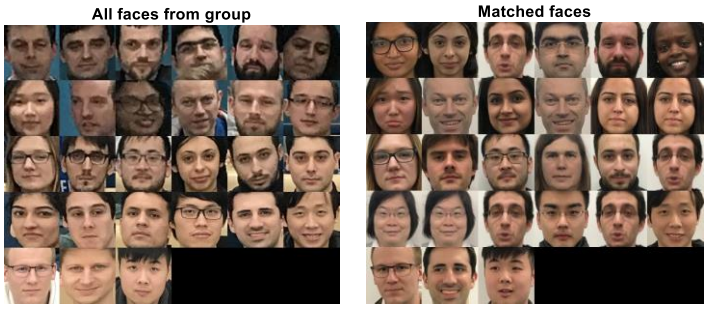


Figure 3: visualisation of faces in group photo and matched faces using classifier only trained on single portrait image library shows poor performance (43%).

The high accuracy achieved on validation was therefore interpreted as the models over-fitting the training data. On reflection, this also reflected the image augmentation strategy which meant that test and training data from the same original image was being mixed between training and validation sets. However, the fact that test performance was worse on group photos than unseen portrait photos meant that a new approach was required beyond just a stricter test strategy.

Stage 3: Extension of face image library with group photos and image masking

Faces extracted from group photos were visibly different to those extracted from single portraits. They were often more blurred (particularly for those students at the back of the room), had wider variations in illumination and different backgrounds.

Two approaches were therefore adopted to allow for this variation:

1. Faces extracted from group photos were added to training data. The automated face extraction process with the v1 CNN classifier were used as a first pass. Any misclassified images were corrected manually.
2. Adopting an approach proposed for face ID systems (Mohammad, Mohammad, & Alkhatib, 2016), images were converted to grayscale and a mask filter applied to (a) crop 15-pixels from left and right and (b) add a v-shape filter around the neck-line as shown in figure 4.

Figure 4: masking to remove background information

Stage 4: Refined test strategy

A new end-to-end test strategy was adopted. Testing was focused on the end-to-end RecogniseFace script and less emphasis placed on validation at the model-building stage. Secondly, a clear separation of training and test data was achieved by assigning a number of images to a test set before image augmentation (one portrait image per student, approximately 10 video frames per student and 3 group photos). This ensured that images derived from the same original image could not exist in both training and test data.

Image augmentation was performed on the training data as described in stage 2, resulting in the data sets summarised in table 2.

	Training images (before augmentation)	Augmentation	Training (after augmentation)	Test images
Images	740	9	6660	53
Video	1379	9	12411	459
Group extracts	1308	6	7848	3 whole group photos ~ 70 individuals
Total	**3428**		**26919**	**~ 580 faces**

Table 2: training / test data summary

Training data was split 90%:10% for training/validation iterations of each model before running on the test data set.

2.4 Face Recognition – final approach

Table 3 summarises the final approach to face recognition, reflecting the evolution already described in this report.

Pre-processing	Approach	Details
Training Images	100x100 grayscale with background removal mask applied	
Augmentation	Image/video frames – 9x	Scale/rescale% 40%, 65%, 100% Brightness% 70%,100%, 130%
Augmentation	Group photos – 6x	Scale/rescale% 80%, 100% Brightness% 70%,100%, 130%

Model	Approach	Details
MLP	1 fully connected layer size 150	
CNN	2 convolutional layers, 1 fully connected layer size 100 Filter size 4x4, number of filters 32	
CNX (CNN with transfer learning from Alex net)	Alex net convolutional layers plus one fully connected layer	
SVM	Linear kernel function	

Feature extraction	Approach	Details
HOG features	Cell size 8x8 nBins 6 or 9	6 bins for SVM 9 bins for MLP
Bag of features	Grid step 8x8 or 10x10 Bag of words 1000	Grid step 10x10 for SVM Grid step 8x8 for MLP

Table 3: face recognition final approach

Results of face recognition are presented together with emotion classification in section 0.

2.5 EMOTION – OVERVIEW

To address the optional emotion recognition task, an existing dataset of faces with labelled emotions (Goodfellow et al., 2013) was downloaded from Kaggle ("Challenges in Representation Learning: Facial Expression Recognition Challenge | Kaggle," n.d.). The data consists of 48x48 pixel grayscale images of faces categorised based on facial expression into one of seven categories (Angry, Disgust, Fear, Happy, Sad, Surprise, Neutral).

The coursework project required classification into 0=Happy, 1=Sad, 2=Surprised and 3=Angry therefore 5,000 images for each of these four categories was extracted from the Kaggle dataset, resized to 100x100 pixels and saved to jpg format for processing with Matlab's imageDatastore function.

A single CNN emotion image classifier was then developed using this data and an 80%/20% train/validation split before testing on the same test student data set as used for face recognition.

2.6 EMOTION – FINAL APPROACH

The final approach to emotion recognition was as summarised in table 4.

Pre-processing	Approach	Details
Training Images	100x100 grayscale, 5,000 of each category	
Augmentation	No augmentation as sufficient data available	

Model	Approach	Details
CNN	4 convolutional layers, 1 fully connected layer size 100 Filter size 7x7, number of filters 64	

Table 4: emotion recognition final approach

2.7 SUMMARY – FLOW DIAGRAM (FACE DETECTION AND FACE/EMOTION CLASSIFICATION COMBINED)

The final approach to model training, face detection and face/emotion recognition is shown in figure 5.

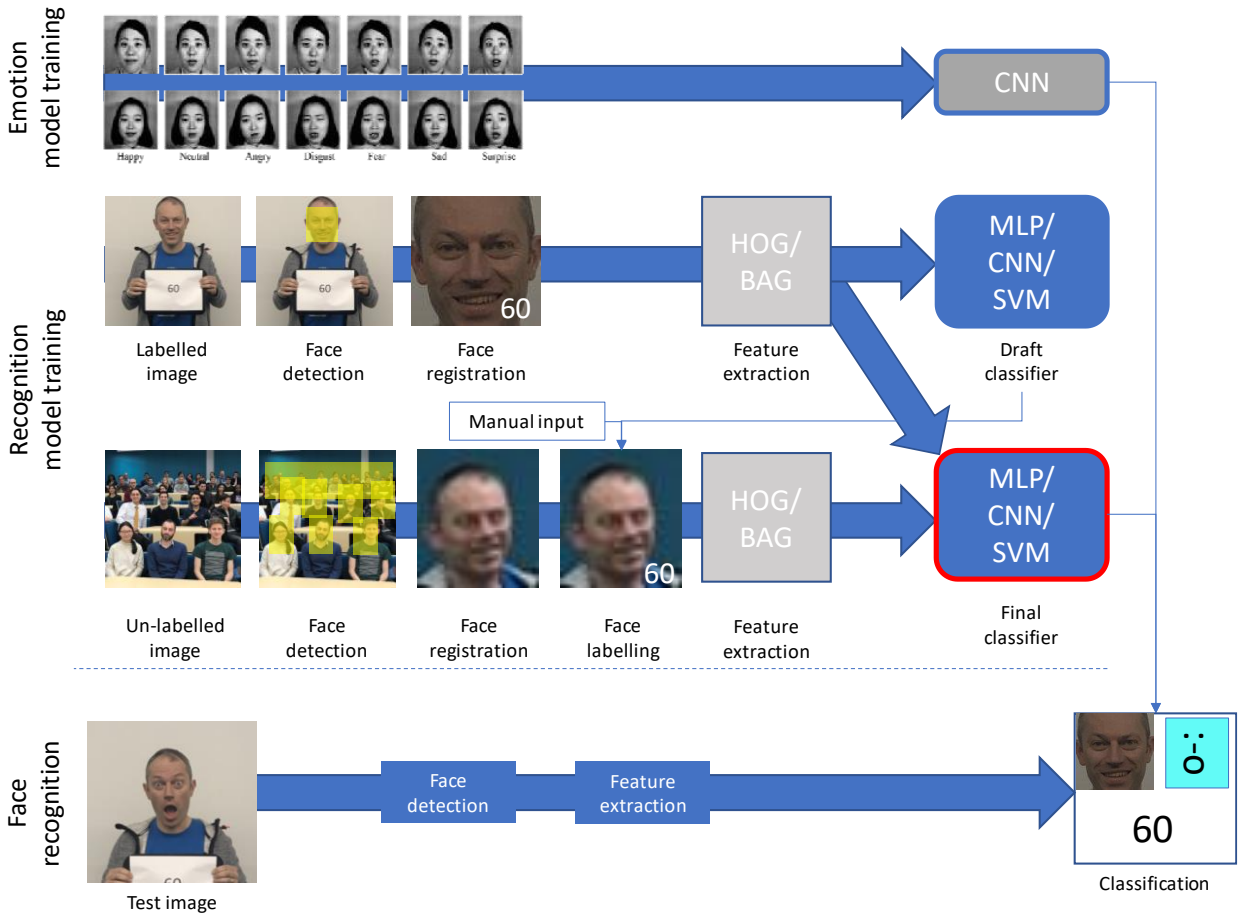


Figure 5: flow diagram for model training, face detection and face/emotion recognition

2.8 Face and Emotion Recognition– results analysis

2.8.1 Results Overview: Visualisation of results

To facilitate testing of the face and emotion recognition function, a visualisation script was developed that displayed results from either (a) batches of image or video input files or (b) individual group photos.

Visualisation of results of face detection, face classification and emotion classification from a sample of 100 labelled test images is shown in figure 6.

Figure 6: visualisation of classification results on 100 test images together with explanatory legend below.

In this example, the classifier was an MLP with feature extraction using Bag of Features

Visualisation of results from a group photo are shown in figure 7. Here, images are unlabelled therefore predicted labels need to be checked against by eye. The visualisation shows a ground truth image from predicted face ID together with emotion.

Figure 7: visualisation of classification results on a test group image together with explanatory legend below.

2.8.2 Results Overview: Emotion Recognition

As there are no 'ground truth' labels for the correct emotion label of student face images, definitive accuracy measures were not calculated. Emotion classification performance was therefore judged subjectively. Based on results such as those shown in figure 6, accuracy was judged to be 80-85%.

Difficult-to-classify images (occluded, odd directions or blurred) appeared to default to 'happy' – obviously a good philosophy on life! Note that a 'neutral' class label was available from the original emotion image dataset but not included in training as outside the defined scope. Including this class could have addressed the 'neutral = happy' observation more appropriately.

Figure 8: examples of 'correct' emotion / face classification (left) and correct identity but 'incorrect' emotion classification (below)

'Cross' label = maybe happy? Unhappy label = maybe 'surprised'?

2.8.3 Results Overview: Face recognition on single face images

Correct face detection and correct face recognition are both required for face recognition to be judged a success. As a result of the process flow adopted (see section 2.7), the choice of classifier impacts face detection (by virtue of the 'not a face' class) as well as face recognition. As both stages are dependent on the choice of classifier, both were considered in assessing classifier performance.

Test results by classifier on the 512 labelled test images were as shown in table 5. Examples of errors are shown in figure 9.

Model	Feature extraction	Images returning multiple detected 'faces'	Images failing to return a detected face	Images with one face detected but miss-classified(1)	Overall test accuracy
MLP	HOG	2/512	16/512	6/512	95.3% << Best
	Bag of features	2/512	14/512	14/512	94.1%
	Raw pixels	3/512	13/512	13/512	94.3%
CNN	N/a	5/512	14/512	16/512	93.2%
CNX (AlexNet)	N/a	7/512	14/512	13/512	93.4%
SVM	HOG	3/512	17/512	10/512	94.3%
	Bag of features	9/512	12/512	38/512	88.1% << Worst

Table 5: Face classification results on 512 test images. Note 1: Includes type (c) and type (d) errors shown in figure 9.

Figure 9: examples of face recognition errors.

Left to right:

(a) Two image regions detected as faces
(b) No face detected (blurred image)
(c) One image region detected but incorrectly
(d) Correct image region detected but classified incorrectly

The best classification accuracy achieved on the 512 test images was 95.3% from the MLP with HOG feature extraction. Errors (in total 4.7%) included 3.5% face extraction errors (no face or multiple faces) and 1.2% classification errors.

2.8.4 Results Overview: Face recognition on other test images

Group images

The RecogniseFace function was tested on the single held-out group test photo. Results on this image are shown in table 6.

Model	Feature extraction	Images failing to return a detected face	Images with one face detected but miss-classified	Overall test accuracy
MLP	HOG	0/26	2/26	92%
	Bag of features	0/26	7/26	73%
	Raw pixels	0/26	3/26	88%
CNN	N/a	0/26	2/26	92%
CNX (AlexNet)	N/a	0/26	1/26	96% << Best
SVM	HOG	0/26	1/26	96% << Best
	Bag of features	0/26	11/26	58% << Worst

Table 6: Face classification results on 1 group image containing 26 students

Non-face images

The RecogniseFace function was also tested on a small sample of images that do not include human faces. With MLP/HOG classifier, as shown in figure 9, one in nine images returned a false positive. The same performance was achieved with other classifiers except for CNX and SVM/Bag of Features which found two false positives: the face-like feature in the tower building and the bear (but not Noddy!). Note that an emotion response of 0 is always returned even if no face detected.

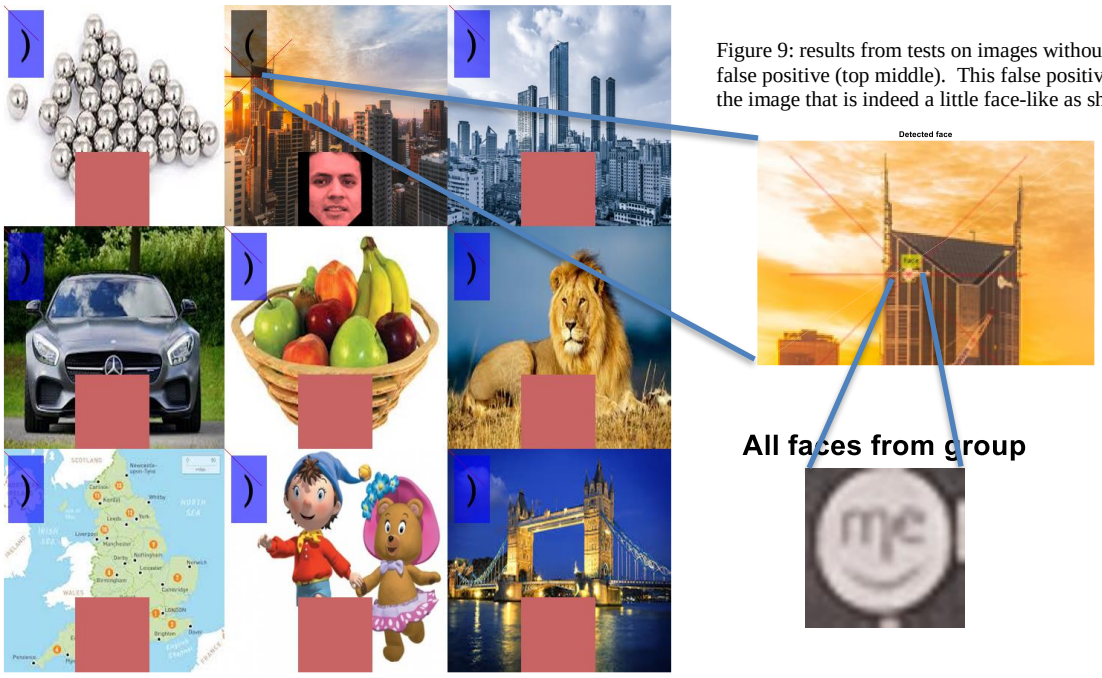


Figure 9: results from tests on images without human faces showing one false positive (top middle). This false positive comes from a small area of the image that is indeed a little face-like as shown below.

2.8.5 Best performing classifier (MLP/HOG) and confusion matrix

Overall, the best performing changed depending on the test set. MLP/HOG was slightly better on single portraits and non-faces and CNX (the Alex Net CNN classifier) was slightly better on group photos. As MLP/HOG performance was better on two out of three tests, this was selected as the preferred model.

A confusion matrix for MLP/HOG classifier on the 512 single-face evaluation dataset is presented in figure 10.

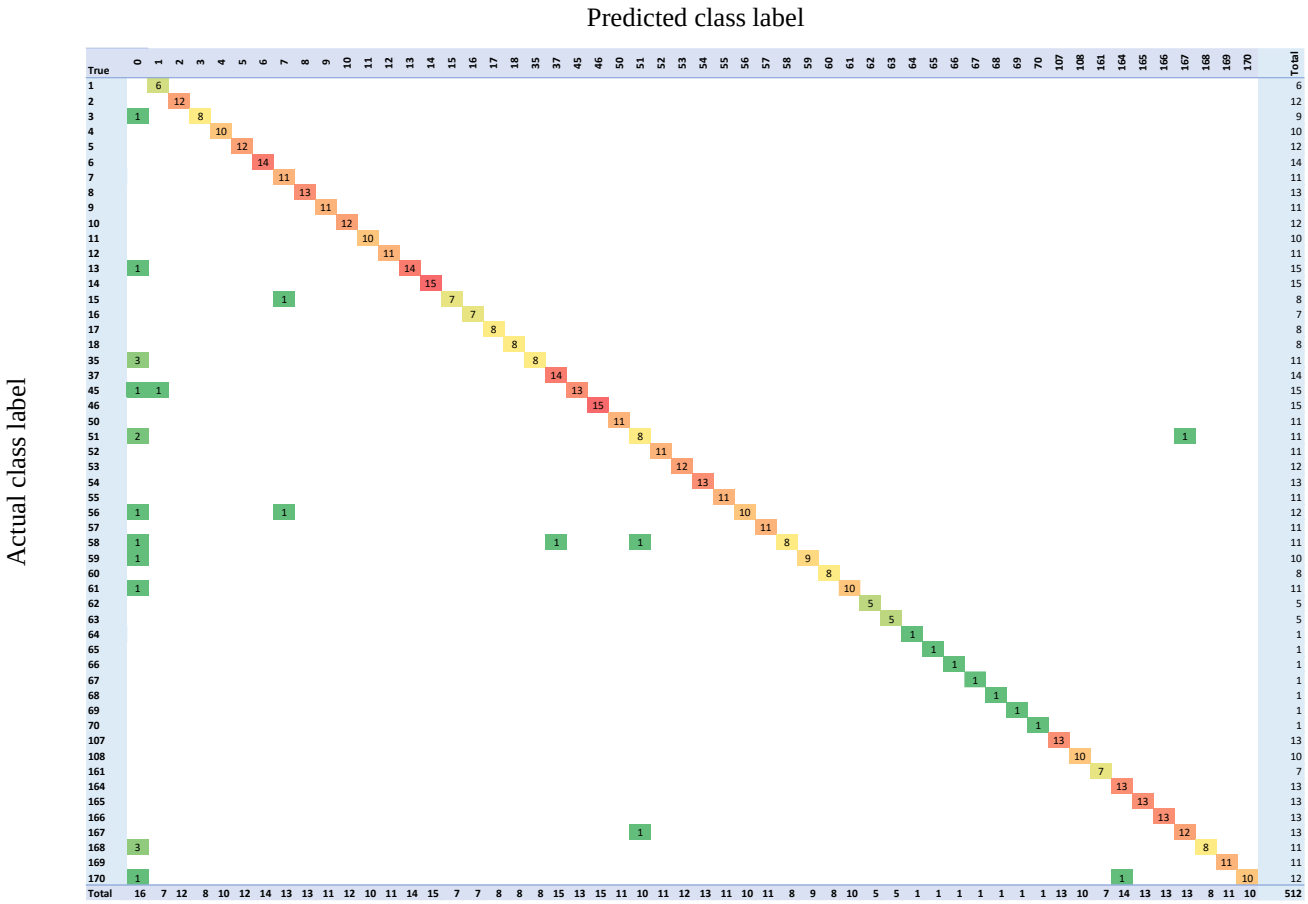


Figure 10: Confusion matrix for MLP/HOG classifier on 512 faces in evaluation set.

2.9 FACE AND EMOTION RECOGNITION– FUNCTION CALL

RecogniseFace - detect and reocgnise face(s) and emotion within RGB image

Syntax and input arguments

P = RecogniseFace(I, featureType, classifierName) – detect and reocgnise face(s) and emotion within RGB image

P = RecogniseFace (I, featureType, classifierName, verbose) – with additional diagnostics if verbose = true

featureType/classifierName parameters:

featureType	classifierName	Note
	CNN	Convolutional neural network, featureType ignored
	CNX	Convolutional neural network, created by transfer learning from Alex Net, featureType ignored
HOG	MLP	Multi-layer perceptron using HOG feature extraction
BAG	MLP	Multi-layer perceptron using Bag of Features feature extraction
RAW	MLP	Multi-layer perception using raw image pixel input (no feature extraction)
HOG	SVM	SVM using HOG feature extraction
BAG	SVM	SVM using Bag of Features feature extraction

Output Arguments

P - Nx4 cell array where N is the number of people detected in image I. The four columns represent:

1. id, a unique number associated with each person that matches the number in the training database provided. If faces are identified as one of those found in training from group photos but without a set of labelled images, id in column 1 is given as a 3-digit number 99x.

2. x, the x location of the person detected in the image (central face region)

3. y, the y location of the person detected in the image (central face region)

4. the person's emotional state, using the following labels: happy = 0; sad =1; surprised = 2; angry = 3.

P - [ ]. If no faces are found, the function returns an empty array.

3 OCR

3.1 OCR – OVERVIEW

Initial experimentation with Matlab's 'ocr' function quickly showed number detection to be reliable given an image consisting only of text but not reliable with our labelled images that included people and background content.

Most of the development effort for this section of the project was therefore concentrated on developing a process for identifying text regions of interest most likely to contain relevant text information, before applying OCR.

A number of different strategies were attempted, with the most encouraging initial results coming from an approach involving Maximally Stable Extremal Region detection with 'detectMSERfeatures' followed by filtering and amalgamation of regions as proposed by the Matlab tutorial ("Automatically Detect and Recognize Text in Natural Images - MATLAB & Simulink," n.d.).

However, this approach produced too many false positive results therefore the region filtering logic needed further adaptation. Using the fact that text in our image library always appeared on a clean white background, regions of interest were filtered based on the colour of their rectangular perimeter, as further described below.

The same approach was followed for video images, after first extracting sample video frames. Blurred images were found to be problematic but basic image sharpening techniques caused no noticeable improvement therefore the OCR process was not further refined.

3.2 OCR – FINAL APPROACH

Each image was rescaled to 1,008 pixels high to ensure a consistent input size then MSER features extracted with size 20-500 and stepsize for intensity threshold 4. Region filtering was then performed as follows:

- Bounding boxes increased by a 15-pixel margin on all sides
- Overlapping boxes merged
- Mean perimeter colour calculated for each bounding box
- Threshold colour set at lower of 210 or brightest box found (so we always keep at least one box)
- Any bounding boxes with perimeter colour below the threshold rejected
- OCR performed
- If more than one text result, reject any text detected as white space or where OCR confidence < 75%.

3.3 OCR – FLOW DIAGRAM

The OCR detection process is illustrated below in figure 11.

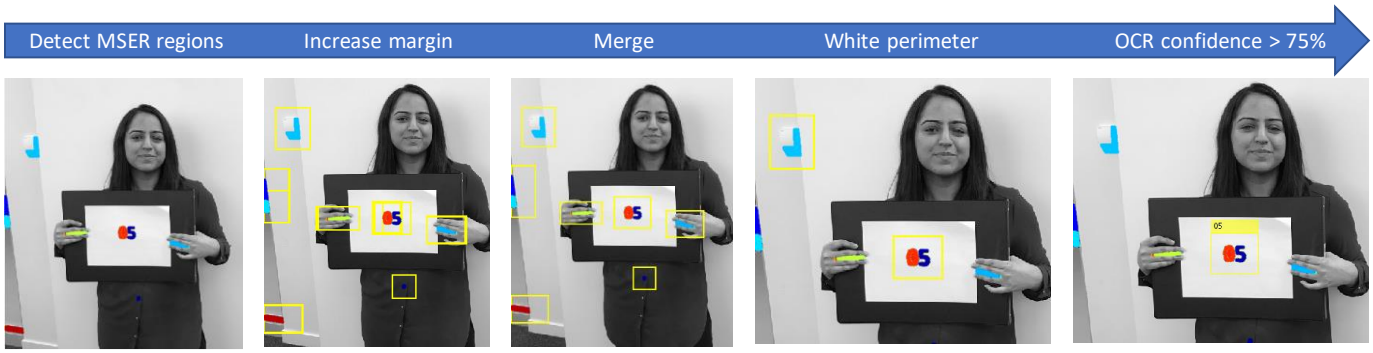


Figure 11: OCR flow diagram

For video processing, the OCR function extracts one frame every 0.5 seconds within the entire video duration, processes this as above for a stationary image, removes nul or white space characters and returns both (a) a majority vote across all sampled frames and (b) all frame-by-frame results.

3.4 OCR – RESULTS ANALYSIS

Results of OCR function processing on 512 test images (including some from stills and video) are summarised in table 7.

	Results	Accuracy	Comment
Test still images	53/53 correct	100%	
Test single video frame images	406/449 correct	90%	Lower accuracy mostly caused by slightly blurry images
Test video files	10/10 correct	100%	5/10 correct in all sampled videoframes. Remainder correct based on a majority vote.
Test multiple OCR regions (2 or 3 combined images)	2/3 correct	67%	Only 1 of 2 OCR regions detected in one file, due to slightly lower lighting resulting in white perimeter being rejected.

Table 7: OCR results on evaluation data

Note: accuracy on group images could be increased by lowering the 'white perimeter' colour threshold from 210 to 205 but no change was made due to the risk of a negative impact on performance for other images.

3.5 OCR – FUNCTION CALL

detectnum - detect one or more OCR regions on clean white region of busy image or video file

Syntax and input arguments

ocrResult = detectnum(I) – detect OCR regions in .jpg image or .mov video file

ocrResult = detectnum(I, verbose) – with additional diagnostics if verbose is specified as integer 1, 2 or 3

(1 being the least and 3 being the most diagnostic information)

Output Arguments

ocrResult - Nx1 or 1x2 cell array

If I is an image, ocrResult is Nx1 cell array where N is the number of positive OCR responses detected.

If I is a video, ocrResult is 1x2 cell array containing (a) single 'majority vote' response and (b) structure array containing Nx1 cell array being single OCR result from N video frames sampled every 0.5 seconds from I.

If no positive OCR responses, ocrResult (or individual element of structure array) provides the following error code:

- 'Z' if no MSER regions found; and
- 'X' if MSER regions found but no text detected.

4 Scripts and Files Provided With This Report

The core function files for this coursework are the following Matlab files:

1. detectnum.m
2. RecogniseFace.m

Details of the inputs to and outputs from these functions are provided in the project report, sections 2.9 and 3.5.

In addition, the following files are provided:

File/folder type	**File/folder name**	**Contents**
Txt file	_readme.txt	Intro file
Matlab file	detectnum_TestHarness_images.m	Test harness for detectnum, operates on still image file folders
Matlab file	detectnum_TestHarness_videos.m	Test harness for detectnum, operates on video file folders
Matlab file	RecogniseFace_TestHarness.m	Test harness for RecogniseFace, operates on still image file folders
Folder of images	facesRef	Reference images for each student class
Folder of classifier models	models	
Folder of images	testimagesGp	Held out group images
Folder of images	testimagesIV	Held out portrait images (extracted from both Images and Videos)
Folder of images	testimagesMultiOCR	Test OCR images containing multiple OCR regions
Folder of images	testimagesNoface	Test images with no faces present
Folder of images	testimagesVFull	Held out test videos

5 DISCUSSION

This project has developed a function for both face/emotion recognition and OCR. The development of individual models was found to be relatively straight forward in MATLAB. However, after much trial-and-error, it became clear that the most challenging part of the project was to bring everything together to create classifiers that were reliable given unseen images that vary in size, quality and background content. Visualisation tools were found to be an important part of experimentation, testing and overall improvement.

Different elements of the project are considered further as follows:

Face detection

Strengths	Combining the face detector function with a negative class in the face classifier resulted in a good balance between minimising false negatives and false positives.
Weaknesses	The 'not-a-face' discriminating class was not always reliable, and several instances occurred where clothing was detected as a face. Due to the approach adopted to cope with detection of multiple bounding boxes, false positives are more likely if there is no face present in the input image.
Future work	Detection could be improved by increasing the number of images used to train the not-a-face category. An intermediate binary classifier (face / not-face) could be introduced before the multi-class face ID classifier.

Face classification

Strengths	Including some images extracted from group photos in training and adopting a mask filter to minimise background noise made a significant improvement to accuracy. Visualisation of errors was very helpful in project evolution. Good accuracy, particularly with MLP/HOG and CNNs.
Weaknesses	Requirement for semi-manual classification of group images was as inefficient stage in the project. Classifiers utilising bag of features did not perform as well as expected (especially SVM) and may not have been optimised fully.
Future work	The following could be investigated to see if accuracy rates could be improved: - Greater levels of image augmentation to create a larger dataset, including techniques such as rotation - Further optimization of individual classifiers - Integration of an SVM classifier on top of features extracted from Alex Net convolutional network - Inclusion of classification probabilities into workflow. For example, two instances of the same person identified in one photo means that one is very likely an error. - Aggregation of different models into an ensemble classifier, for example by majority vote

Emotion

Strengths	Reasonably good performance from initial CNN optimization, perhaps because dataset was significantly larger than face classifier database.
Weaknesses	No ground truth to calculate accuracy figures. Only one model (CNN) developed therefore no comparison performed.
Future work	A neutral class would be useful in order not to return an arbitrary emotion for a neutral expression. Pre-classification of test images would allow an objective calculation of emotion classifier accuracy. It would be worthwhile investigating the following to see if accuracy can be improved: - Adopting the masking approach used in face classification to remove impact of background in emotion images - Training CNN by transfer learning from Alex Net rather than from scratch

OCR

Strengths	MSER with white perimeter worked well to find regions that OCR can cope with.
Weaknesses	White perimeter approach may reject one region of interest if two are found with different brightness.
Future work	Investigate regional-CNN. Investigate active contour ('activecontour') or image erosion ('imerode') as alternatives to MSER. Train detection classifier on the specific font being detected.

6 References

Automatically Detect and Recognize Text in Natural Images - MATLAB & Simulink. (n.d.). Retrieved March 20, 2018, from https://uk.mathworks.com/help/vision/examples/automatically-detect-and-recognize-text-in-natural-images.html?requestedDomain=true

Challenges in Representation Learning: Facial Expression Recognition Challenge | Kaggle. (n.d.). Retrieved March 19, 2018, from https://www.kaggle.com/c/challenges-in-representation-learning-facial-expression-recognition-challenge/data

Face Recognition - MATLAB & Simulink. (n.d.). Retrieved March 20, 2018, from https://uk.mathworks.com/discovery/face-recognition.html

Fawzi, A., Samulowitz, H., Turaga, D., & Frossard, P. (2016). Adaptive data augmentation for image classification. In *2016 IEEE International Conference on Image Processing (ICIP)* (pp. 3688–3692). IEEE. https://doi.org/10.1109/ICIP.2016.7533048

Goodfellow, I. J., Erhan, D., Carrier, P. L., Courville, A., Mirza, M., Hamner, B., … Bengio, Y. (2013). Challenges in Representation Learning: A Report on Three Machine Learning Contests (pp. 117–124). Springer, Berlin, Heidelberg. https://doi.org/10.1007/978-3-642-42051-1_16

Mohammad, A. H., Mohammad, B., & Alkhatib, B. (2016). Websites Authentication Based on Face Recognition. *Asian Journal of Information Technology*. Retrieved from https://pedia.svuonline.org/pluginfile.php/53/mod_resource/content/1/Websites Authentication Based on Face Recognition.pdf

CV class of 2018. What a '0=happy' class!