Hidden and Face-like Object Detection using Deep Learning Techniques – An Empirical Study

Anirudh A[1]
[1]Computer Science and Information Systems
BITS Pilani
Hyderabad, India
f20180936@hyderabad.bits-pilani.ac.in

Subrata Chakraborty[2, 3]
[2]School of Science and Technology,
Faculty of Science, Agriculture, Business and Law, University of New England, Armidale, NSW, 2351, Australia
[3]Centre for Advanced Modelling and Geospatial Information Systems (CAMGIS), Faculty of Engineer and IT, University of Technology Sydney, Sydney, NSW, 2007, Australia
subrata.chakraborty@une.edu.au

***Abstract* – An essential aspect of artificial intelligence is how closely machines can mimic humans. One of the motivations for developing intelligent systems is human vision. While trying to recognise a class of images, it is as vital to distinguish the class of images from similar-looking objects and identify them in hidden places as it is to create bounding boxes and learn to localize the position of the object. Traditionally, deep learning models have performed exceptionally well in image classification and object detection tasks. In this work, we perform four experiments to train machines to distinguish between real faces and face-like objects and to recognise them. Nine state-of-the-art deep learning-based classifiers have been chosen to perform a comparative study on the designed experiments. Using these experiments, we establish that training models on real faces does not prepare them to identify face-like objects, and at the same time, training on face-like objects enables the models to detect face-like images even while hidden amongst other images. Despite work being done in the fields of camouflage detection and optical illusion detection, to the best of our knowledge, no work has been done in training and testing machines to distinguish between face and face-like objects with deep learning methods. This work could help researchers make better camouflage detection systems, perform context sensitive studies, understand the biases that various models possess towards certain classes of images, and have applications in real life such as military and self-driving cars.**

***Key Words* - Object detection, image classification, data augmentation, vanishing gradient, dying ReLU, face-like objects.**

I. INTRODUCTION

The human brain is a highly complex organ and is in fact, regarded by many as the most complex thing discovered in the universe. To humans, vision seems easy and trivial, but in reality, the human brain processes roughly 65 images every second with millions of pixels in each image. Despite its complexity and ability in face recognition and discrimination, research suggests that the brain may not have developed a specific potential to understand faces. It has been proven [1] that to distinguish between faces, the brain uses the same kind of pattern recognition as it uses to differentiate between objects such as plants, animals, and cars. Since the goal of artificial intelligence (AI) includes mimicking human cognitive activities such as perception, reasoning, and learning, it becomes interesting to understand if machines can distinguish between real faces and face-like objects and further identify them when hidden between other images (or the background). Recognising hidden images and differentiating between similar objects are challenging tasks for the human brain itself. Hence, we try to understand how various deep learning techniques (which have performed exceptionally well in image classification) can perform in such tasks and what sort of training is required for the same. Such empirical studies could become useful in practice, such as in military applications to uncover camouflage and self-driving vehicles where it is crucial to distinguish between face and face-like objects and at the same time recognise any hidden face or face-like objects.

Deep learning techniques have had several applications such as COVID-19 detection through transfer learning [2], medical image analysis [3], image segmentation [4], autonomous driving [5], etc., in the field of computer vision. Deep learning techniques are found to be performing exceptionally well in object detection and image classification [6, 7]. They are also found to be suitable for camouflage detection tasks [8, 9]. Further, since classifiers built using deep learning models outperform traditional machine learning based classifiers in classification problems [10, 11], the use of models such as k-nearest neighbours and support vector machine has been avoided.

In this study, we perform an empirical assessment of nine state-of-the-art deep learning classifiers including a custom-made Deep CNN model (Fig. 1), MobileVNet [12], Single Shot Detector (SSD) [13], AlexNet [14], InceptionV3 [15], VGG19 [16], ResNet50 [17], GoogleNet [18], and DenseNet121 [19]. All the deep learning image classification techniques use convolutional neural networks [20], which in turn make use of 4 types of layers – convolutional (conv) layers, rectified linear unit (ReLU) layers, pooling layers, and fully connected (or dense) layers.

The empirical study consists of four distinct experiments (three classification problems and one object detection problem) to assess the comparative performance of the selected deep learning models. The first experiment is a classification problem where the predictive ability of the nine classifiers is tested in distinguishing between face and face-like objects. To demonstrate the importance of training the models on face-like objects, the second experiment trains the models on two classes – real faces and food items. The testing is done on food items in the shape of faces to understand what features are being learnt and what sort of training is necessary. In the third experiment, the classifiers

are tested on images where faces are hidden between trees to understand the predictors' confidence and their biases towards one particular class (if any). Further, the fourth experiment is performed to study how machines can locate hidden images, small objects, and multiple objects. Object detection involves two steps – (i) finding an arbitrary number of objects and (ii) classifying every object along with bounding boxes. The detectors are said to be single-stage or two-stage models [21] based on whether the two steps are performed together or separately. In this work, YOLO V3 [22], a single-stage object detector, has been used to detect images.

In the following sections, we discuss the related literary evidence followed by the dataset. Then, we explain the experimental setup followed by the results and discussions.

II. RELATED WORK

This section highlights the different studies available in the broad field of classifying and recognising hidden and confusing images. An exhaustive review of various state-of-the-art face detection studies is done in [23]. The paper also enlists various challenges and applications of face detection and concludes by introducing different standard databases along with their features for face detection.

Hidden image detection could have applications in traffic sign recognition [24]. The authors suggest a learning vector quantization (LVQ) based supervised neural network algorithm to detect traffic signs which seem to be hidden partially because of trees or harsh weather. The key take away is to try and adopt the CNN model, extract different features, use principal component analysis (PCA), and finally use LVQ for classification.

Search Identification Network (SINet) [25] is widely used to tackle the camouflage detection problem. The paper uses a densely annotated dataset (COD10K) consisting of 10,000 images across 78 categories involving camouflaged objects. The framework outperforms 12 cutting edge baselines [26] on all tested datasets. Further the scope of decamouflaging to identify foreground objects hidden in the background image is discussed. The target object is revealed by discriminating the foreground object from the camouflaged image. The work has applications in salient object detection [27] as well, where the primary goal is to train machines to identify the most significant object(s) in any given image.

Significant work [28, 29] has been done in detecting optical illusions which conclusively prove that AI and CNNs are deceived by optical illusions just like humans. It has been shown that CNNs trained on natural images react like humans when tested on colour-based (visual) optical illusions. Similarly, models trained on actual motion videos get deceived and recognise motion-like images (illusions) to be videos. They in fact predict the direction of motion. It has been debated [30] if CNNs should actually reproduce optical illusions. The key take away however has been that one could make much better use of DNNs by focussing more on their differences rather than similarities humans.

Despite various studies being done in the fields of camouflage detection and optical illusion detection, to the best of our knowledge, no study discusses the ability of machines to differentiate between face and face-like objects, recognise them, and further understand the inbuilt biases that various models show towards certain classes.

III. DATASET

The datasets have been handpicked very carefully to choose images that closely resemble human faces. The training data [31, 32, 33, 34] used in this work can be categorised into five groups – D1 (real faces – 300 images), D2 (black and white sketches of faces – 250 images), D3 (face-like objects – 290 images), D4 (food items – 300 images), and D5 (sketches as well as images of trees – 300 images). Special care has been taken to ensure that the dataset D4 does not contain any image that resembles face-like objects. D3 has been used specifically to show that AI models can perform significantly better in recognising face-like images and human-like objects when trained extensively on such images rather than on real human faces. The test data can likewise be categorised as T1 (real faces – 30 images), T2 (face-like objects – 25 images), T3 (food items that appear similar to faces – 40 images), and T4 (trees that have face-like objects merged in them – 33 images). "Fig. 2." and "Fig. 3." show examples of images from each category of the training and testing datasets. For each experiment, a suitable subset of this dataset is chosen.

IV. EXPERIMENTAL SETUP AND STUDY DESIGN

In this work, four experiments are conducted to study the performance of various image classification and object detection techniques in recognising confusing and hidden objects. While YOLO V3 [22] is used for object detection, nine state-of-the-art deep learning classifiers - deep CNN (Fig. 1), MobileVNet [12], Single Shot Detector (SSD) [13], AlexNet [14], InceptionV3 [15], VGG19 [16], ResNet50 [17], GoogleNet [18], and DenseNet121 [19] are studied in the first three experiments. The custom-made deep-CNN model makes use of two Conv2D layers (each followed by a MaxPooling2D layer), another Conv2D layer, a Flatten layer, and two Dense layers. Using accuracy and F1 scores, the performance of the models is compared.

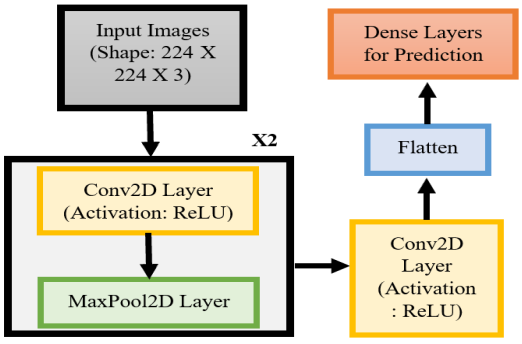

Fig. 1. Deep CNN Model

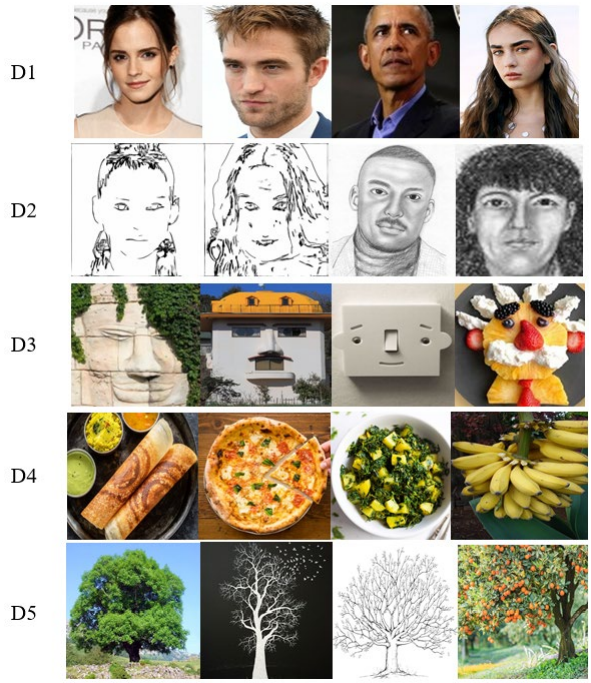

Fig. 2. Training Data - Sample Images

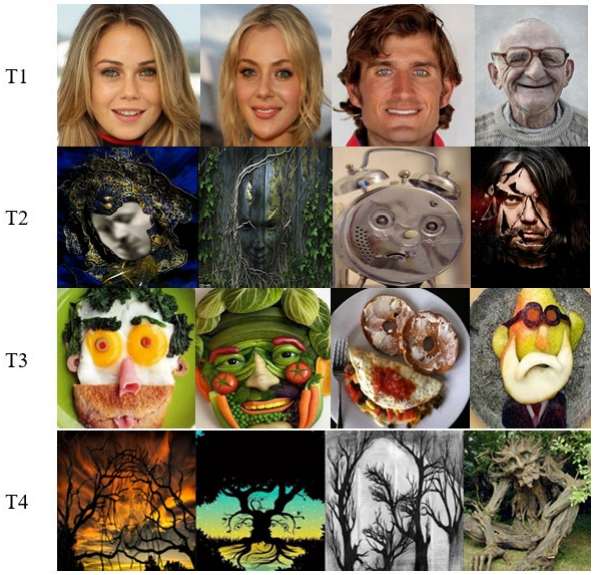

Fig. 3. Testing Data - Sample Images

To maintain uniformity, all nine models are trained using ImageNet data, compiled using sparse categorical cross entropy function and the Adam optimizer. In each model, the SoftMax layer has been used to make the final predictions. A learning rate scheduler is used to train the models with multiple training rates (0.01, 0.001, 0.0001, and 0.00001) at different epochs. Image pre-processing includes data augmentation [35] (rotation, random zoom, width/height shift, and horizontal/vertical flip) performed before training to enhance the predictive ability of the models. The summary of the four experiments is depicted pictorially in "Fig. 4" and the following sub-sections explain each experiment in detail:

A. Experiment 1

The experiment aims to study how well image classification techniques can distinguish between face and face-like objects. To understand this, each classifier was trained on datasets D1 and D3 and tested on datasets T1 and T2. Special care was taken to ensure that none of the models encountered the dying ReLU problem [36]. The accuracy and loss values were plotted against the number of epochs, output probabilities were calculated for both classes for each testing example, and the confusion matrix was drawn.

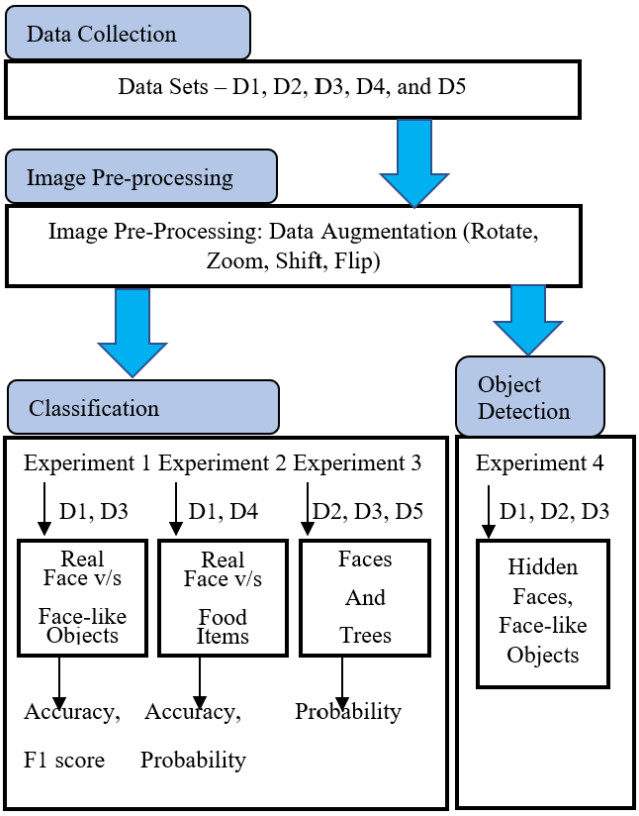

Fig. 4. Experimental Setup

B. Experiment 2

The complexity of the experiment was increased, and the classifiers were now trained on datasets D1 and D4, i.e., no training data had any non-real face-like object in it. The models were trained with real faces as class 1 and food items as class 0. However, the test dataset (T3) contained images of food items in the shape of faces. This was done to understand the importance of training the models on face-like objects in particular. Having ensured that the dataset D4 did not have any images that resembled face-like objects, it was an interesting task to see how the models reacted when tested on T3. The accuracy and loss values were plotted against the number of epochs, output probabilities were calculated for both classes for each testing example, and the confusion matrix was drawn.

C. Experiment 3

The complexity of the experiment was further increased, and the classifiers were trained on datasets D2, D3, and D5. The models were tested on the dataset T5. This

time, since all images had both trees and faces in them, the goal of the experiment was not to classify the images as trees and faces. It was to understand the probability values (confidence of the classifiers) in recognising both trees and faces in the images and, at the same time, understand if the classifiers had a bias towards one of the classes. Hence, rather than testing accuracy and confusion matrix, the metric of interest becomes the SoftMax probability scores.

D. Experiment 4

Having been able to distinguish between face and face-like objects successfully, the object detection experiment is conducted to study how machines can detect face-like objects in the real world. Because of its ability to detect multiple objects, predict classes, identify locations, and locate small objects, YOLO V3 [22] has been chosen to perform this experiment. It has been conclusively proven that the YOLO models outperform older object detection models such as the sliding window technique, different variations of RCNN, etc., because of the use of residual blocks, skip connections, and up sampling. YOLO V3 makes object detection at three different layers (82, 94, and 106). The neural network down samples each image by a factor of 32, 16, and 8, respectively, at these layers, which in turn makes them responsible for detecting large, medium, and small images, respectively. By using convolutional layers of stride 2 (instead of pooling layers) for down sampling, YOLO V3 is able to prevent the loss of low-level features, hence increasing its ability to detect small objects.

The experiment was carried out in three stages – by successively adding more diverse images to the training data to improve the model's predictive ability. First, the model was trained on real faces (a random subset of images from D1) and tested on face-like objects and hidden faces. Then, face-like objects (a random subset of images from D3) were added to the training data in the second phase of the experiment. In the final phase, sketches of faces (a random subset of images from D2) were also included in the training data. The final training data set consisted of 629 images, and the model was trained for 2300 iterations. The test dataset consisted of 50 images where face-like objects were hidden among other objects. For the purpose of this experiment, the last few layers of YOLO V3's transfer learning model were fine-tuned. In particular, the last three YOLO layers (and their preceding convolutional layers) were modified by changing the number of classes and filters to 1 and 18 ((number of classes+5) *3), respectively. The weights computed after training the model for 2300 iterations were saved and later used for testing purposes.

V. RESULTS AND DISCUSSION

Here, we describe and analyse the empirical results and perform a comparative analysis of the various models used. For the sake of a compact representation, the confusion matrix of each model has been represented as a row vector of length 4 – $x_1, x_2, x_3, and\ x_4$ denoting the classes true negative (TN, i.e., rightly classified as class 0), false positive (FP, i.e., wrongly classified as class 0), false negative (FN, i.e., wrongly classified as class 1), and true positive (TP, i.e., rightly classified as class 1) respectively.

A. Results from Experiment 1

The goal of this experiment was to study how well image classification techniques distinguish between face and face-like objects. The models used in this study are trained and evaluated on ImageNet [37], consisting of 1000 classes. "Table 1" describes the empirical results using accuracy, F-1 measure, and confusion matrix. Firstly, "Table 1" clearly indicates that performing data augmentation has a significant improvement in the predictive ability of the models. This is similar to the results achieved by [38]. Further, adding images from dataset D3 (face-like objects) during training has improved the predictive ability of the models.

From "Table 1" we can see that the Deep-CNN model and AlexNet have the lowest predictive ability. This could be because these models are not as dense as the other models, so they struggle to learn complex features from the image sets. A similar justification would hold for MobileVNet [12] where there is a trade-off between small memory size and speed for accuracy. Because of its ability to reuse features and overcome the vanishing gradient problem, DenseNet121 [19] gives the best results. Since every dense block of DenseNet121 is fully connected, the classifier uses features of all complexity levels (unlike other models where only the final layers with high-level features are involved in the prediction) and gives a smooth decision boundary. Although ResNet50 [17] tries to overcome the problem of vanishing gradients using skip connections, the model does not have the same predictive ability as DenseNet121 but performs better than the other models. These results are similar to that obtained by [39]. Understandably, GoogleNet [18] produces better results than Inception V3 [15] since the GoogleNet architecture consists of 9 modules, each module being an Inception block. The use of ReLU layers and $1X1$ convolutional layers help VGG19 [16] classify the images into real and face-like images successfully by making the decision function more non-linear and at the same time not changing the respective fields. Initially, VGG19 predicted that all images belonged to the same class (class 0). Adjusting the learning rate appropriately helped the model overcome the Dying-ReLU problem, improving its predictive ability. The results thus prove that training the models using real faces as well as face-like objects allows the models to distinguish clearly between the two classes and recognise each class distinctly.

B. Results from Experiment 2

To show the importance of training on face-like objects, in experiment 2, the classifiers were trained on datasets D1 and D4, i.e., no training data had any non-real face-like object in it. The models were trained with food items as class 0 and real faces as class 1. The test dataset (T3) consisted of only food items, i.e., class 0 (no images from

TABLE 1: Empirical Results - Experiment 1

	Before Data Augmentation (Trained with D1)			After Data Augmentation (Trained with D1, D3)		
	Accuracy	*F1-Values*	*Confusion Matrix (TN, FP, FN, TP)*	*Accuracy*	*F1-Values*	*Confusion Matrix (TN, FP, FN, TP)*
Deep CNN	0.527	0.235	25, 0, 26, 4	0.745	0.787	15, 10, 4, 26
MobileVNet	0.69	0.78	8, 17, 0, 30	0.8	0.825	18, 7, 4, 26
SSD	0.636	0.743	6, 19, 1, 29	0.82	0.857	15, 10, 0, 30
AlexNet	0.69	0.779	8, 17, 0, 30	0.83	0.84	15, 10, 0, 30
InceptionV3	0.636	0.743	6, 19, 1, 29	0.85	0.883	17, 8, 0, 30
VGG19	0.545	0.7	0, 25, 0, 30	0.85	0.875	19, 6, 2, 28
ResNet50	0.745	0.787	15, 10, 4, 26	0.91	0.923	20, 5, 0, 30
GoogleNet	0.85	0.88	17, 8, 0, 30	0.91	0.93	21, 4, 1, 29
DenseNet121	0.781	0.833	13, 12, 0, 30	0.963	0.967	23, 2, 0, 30

TABLE 2: Empirical Results - Experiment 2

	Accuracy	Confusion Matrix (TN, FP, FN, TP)
Deep CNN	0.524	21, 19, 0, 0
MobileVNet	0.625	25, 15, 0, 0
SSD	0.949	38, 2, 0, 0
AlexNet	0.725	29, 11, 0, 0
InceptionV3	0.725	29, 11, 0, 0
VGG19	0.949	38, 2, 0, 0
ResNet50	0.975	39, 1, 0, 0
GoogleNet	0.9	36, 4, 0, 0
DenseNet121	1	40, 0, 0, 0

TABLE 3: Probability Values – Experiment 3

	Example-1		Example-2		Example-3	
	Class 0	*Class 1*	*Class 0*	*Class 1*	*Class 0*	*Class 1*
Deep CNN	1.0E+00	3.3E-05	1.0E+00	1.9E-03	9.8E-01	1.6E-02
MobileVNet	9.9E-01	9.3E-03	1.0E+00	7.0E-04	9.7E-01	2.7E-02
SSD	1.0E+00	4.9E-03	1.0E+00	3.9E-05	1.0E+00	3.2E-15
AlexNet	9.5E-01	5.2E-02	1.0E+00	2.4E-03	9.6E-01	4.2E-02
InceptionV3	9.8E-01	1.7E-02	9.3E-01	7.4E-02	9.8E-01	1.6E-02
VGG19	9.3E-01	7.0E-02	9.5E-01	4.8E-02	9.7E-01	3.4E-02
ResNet50	1.0E+00	3.5E-05	1.0E+00	7.0E-07	1.0E+00	4.1E-06
GoogleNet	1.0E+00	2.8E-03	9.9E-01	9.8E-03	1.0E+00	3.3E-04
DenseNet	1.0E+00	3.5E-06	1.0E+00	4.6E-04	1.0E+00	1.2E-05

class 1). However, each of these food items was in the shape of a human face. The test dataset was specifically curated this way to ensure that a high accuracy value is not achieved because of non-confusing, clear images. The results of the experiment are tabulated in "Table 2". Further, the SoftMax probabilities were computed for each classifier and each test image. "Table 3" shows the SoftMax Probability values achieved by each classifier for three randomly picked test images (the same trend can be observed for all images as well). Having no images that resembled face-like objects in D4, all models classified most of the images as food items rather than face or face-like objects. This conclusively proves that the models being deep neural networks understand and learn sophisticated features of the image and not just superficial properties such as shape. The accuracy values follow a similar trend as those in Experiment 1. The F1 scores for all classifiers are 0 since all test images belong to the same class (class 0).

High probability scores for class 0 and negligible probabilities for class 1 indicate that the models have been able to clearly distinguish between the two classes without any overlap, despite the test images being confusing. This suggests that the classifiers are able to extract features of various complexity levels successfully, and it is not sufficient to train the models on real faces in order for them to be able to detect face-like images.

C. Results from Experiment 3

In the third experiment, the classifiers were tested on dataset T5 – images of trees with human-like faces in them. To further prove that training the models on face-like objects is sufficient for the models to recognise such objects (even when hidden), the models were trained on a dataset comprising of images from D2 and D3. Since all images had both trees and faces, the goal of the experiment was not to classify the images as trees and faces. It was to understand the probability values (confidence of the classifiers) in recognising both trees and faces in the images, and at the same time, understand if the classifiers had a bias towards one of the classes. Hence, rather than testing accuracy and confusion matrix, the metric of interest becomes the SoftMax probability scores – samples of which are shown in "Table 4". It can be seen that AlexNet and VGG19 have a bias towards trees while other models recognised face-like objects in majority. ResNet50 is the only model without any such bias. Further, it can be seen that the predictions of VGG19 and ResNet50 are strong, i.e., the recognized class

TABLE 4: Probability Scores (Examples) - Experiment 3

		Example-1	Example-2	Example-3	Example-4	Example-5	Example-6
Deep CNN	Class 0	8.601E-01	3.014E-01	3.588E-01	7.286E-01	4.919E-01	4.493E-01
	Class 1	1.399E-01	6.986E-01	6.412E-01	2.714E-01	5.081E-01	5.507E-01
MobileVNet	Class 0	0.5539197	0.101754	0.5267986	0.758094	0.3020167	0.917007
	Class 1	0.4460802	0.898246	0.4732014	0.241906	0.6979832	0.0829929
SSD	Class 0	0.4919452	0.6514508	0.8788868	0.7826871	0.9782833	0.8169839
	Class 1	0.5080547	0.3485493	0.1211132	0.2173129	0.0217167	0.1830161
AlexNet	Class 0	7.329E-01	5.486E-01	8.601E-01	7.286E-01	4.493E-01	7.756E-01
	Class 1	2.671E-01	4.514E-01	1.399E-01	2.714E-01	5.507E-01	2.244E-01
InceptionV3	Class 0	4.919E-01	6.259E-02	1.721E-01	5.815E-01	9.203E-03	3.016E-01
	Class 1	5.081E-01	9.374E-01	8.279E-01	4.185E-01	9.908E-01	6.984E-01
VGG19	Class 0	1.000E+00	6.711E-22	1.000E+00	1.000E+00	1.000E+00	5.635E-32
	Class 1	6.653E-22	1.000E+00	3.542E-15	6.400E-17	7.151E-10	1.000E+00
ResNet50	Class 0	5.870E-06	9.120E-04	3.701E-06	9.999E-01	9.903E-01	9.999E-01
	Class 1	1.000E+00	9.991E-01	1.000E+00	5.077E-05	9.744E-03	5.300E-05
GoogleNet	Class 0	2.214E-01	7.734E-01	3.759E-01	3.016E-01	4.751E-02	1.787E-02
	Class 1	7.786E-01	2.266E-01	6.241E-01	6.984E-01	9.525E-01	9.821E-01
DenseNet121	Class 0	1.742E-01	3.014E-01	5.990E-01	3.588E-01	1.902E-02	8.675E-01
	Class 1	8.258E-01	6.986E-01	4.010E-01	6.412E-01	9.810E-01	1.325E-01

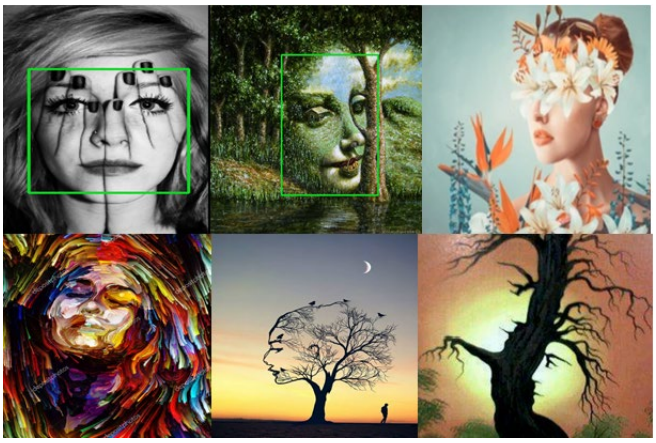

Fig. 5. Object Detection Results After Training on Real Faces

has a probability value of almost 1 while the other class has a negligible value. However, other models predict both classes with non-negligible probability values indicating that both classes are being recognised with almost non-negligible certainty. Hence, it could be concluded that models like ResNet50 and VGG19 can be used when the required predictions have to be highly accurate and certain (for example, applications in the military), while other models can be used when predicting multiple classes is important.

D. Results from Experiment 4

To understand a machine's ability to detect the location of a face-like or hidden object, YOLO V3 was tested on a mix of 50 images from datasets T2, T3, and T4. The training data was successively diversified – initially, all training images were from D1, then images from D3 were added,

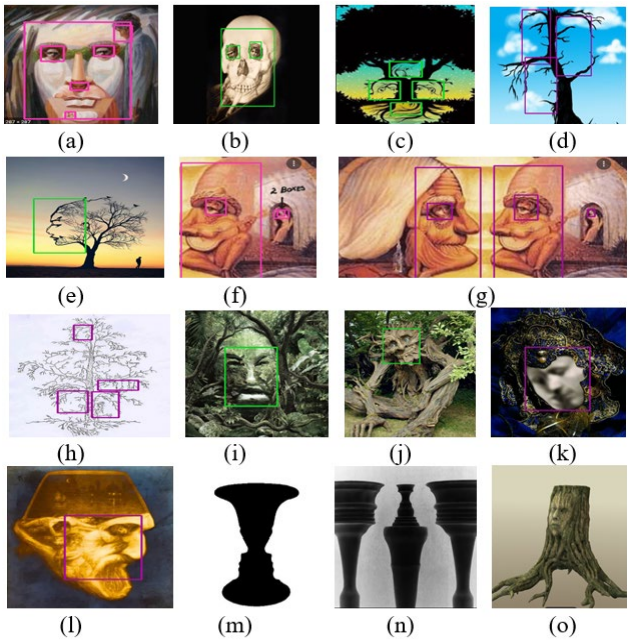

Fig. 6. Object Detection Results After Complete Training

and finally, images from D2 were added to the training dataset. The model was trained for 2300 iterations where an average loss value of 0.103 was achieved. Sample outputs of the experiment have been shown in "Fig. 5." (results after on only D1), and "Fig. 6." (results after complete training). Based on the testing, YOLO V3's capability in object detection can be understood.

Firstly, like the results obtained from Experiments 1, 2,

and 3, the predictions shown in “Fig. 6.” (after training on D1, D2, and D3) are significantly better than those in “Fig. 5.” (after training on only D1). Images a, b, and g of “Fig. 6.” suggest that YOLO V3 can detect small objects with great accuracy because of its ability to decrease the loss of low-level features. Images a, b, c, d, and g of “Fig. 6.” indicate the ability of YOLO V3 to detect multiple objects in the particular, image h of “Fig. 6.” (Satyajit Ray’s [41] famous drawing) indicates the ability of the model to detect face-like objects in highly intricate and hidden sketches as well (however, it is to be noted that not all faces in the drawing have been detected successfully). While images c, d, e, and g in “Fig. 6.” indicate that the orientation of the face is not important for object detection, image k in “Fig. 6.” shows the ability of the model to detect blurred images as well. However, when a face is completely merged with the background (images m and n in “Fig. 6.”), the model is not able to recognise the images indicating that YOLO indicating that it is not possible to directly use YOLO V3 for experiments such as camouflage detection.

The importance of extensive training can be seen by closely examining image f (tested after only 1000 iterations of training) and image g in “Fig. 4”. They suggest that enough training is required for the model to make use of the Intersection Over Union [42] concept to disambiguate between multiple bounding boxes for the same object (an extra box has been obtained in image f and has been pointed using an arrow). Finally, a few cases where YOLO V3 fails to accurately draw bounding boxes are shown in images l, m, n, and o of “Fig. 5”.

VI. CONCLUSION AND FUTURE SCOPE

In this work, we have attempted to understand how various image classification and object detection techniques perform in identifying face-like objects and distinguishing them from real human faces, and to the best of our knowledge, this is this first such study. A comparative study has been done on nine state-of-the-art image classifiers using three experiments. It was concluded that DenseNet121 gives the best results in most cases while AlexNet and MobileVNet perform relatively poorly because of a shallow network. Further, the ability of YOLO v3 to detect small objects, multiple objects, and blurred objects has been tested using an object-detection experiment on a dataset consisting of face-like hidden images. It was also seen that YOLO V3 could not recognise objects when merged with the background completely. This work thus corroborates AI’s ability to mimic humans when it comes to recognising hidden objects and distinguishing between face and face-like objects, provided the model is suitably and sufficiently trained, i.e., it is not sufficient to train the models on just real faces and use them to identify face-like objects. This is a very early stage in such a study. We believe that this work could benefit the research community when similar tests are performed on large scale datasets and tested on practical applications such as self-driving cars and camouflage and hidden object detection for military, where it is crucial to distinguish between face and face-like objects and, at the same time, recognise any hidden face or face-like objects. It is intended to extend this work to perform a context-sensitive study of images for better classification and object detection.

REFERENCES

[1] Young, Andrew W., and A. Mike Burton. "Are we face experts?" *Trends in cognitive sciences* 22.2 (2018): 100-110

[2] Horry, Michael J., et al. "COVID-19 detection through transfer learning using multimodal imaging data." *Ieee Access* 8 (2020): 149808-149824.

[3] Shen, Dinggang, Guorong Wu, and Heung-Il Suk. "Deep learning in medical image analysis." *Annual review of biomedical engineering* 19 (2017): 221.

[4] Minaee, Shervin, et al. "Image segmentation using deep learning: A survey." *IEEE transactions on pattern analysis and machine intelligence* (2021).

[5] Burleigh, Nicholas, Jordan King, and Thomas Bräunl. "Deep learning for autonomous driving." *2019 Digital Image Computing: Techniques and Applications (DICTA)*. IEEE, 2019.

[6] Zhao, Zhong-Qiu, et al. "Object detection with deep learning: A review." *IEEE transactions on neural networks and learning systems* 30.11 (2019): 3212-3232

[7] Papageorgiou, Constantine P., Michael Oren, and Tomaso Poggio. "A general framework for object detection." *Sixth International Conference on Computer Vision (IEEE Cat. No. 98CH36271)*. IEEE, 1998

[8] Bi, Hongbo, et al. "Rethinking Camouflaged Object Detection: Models and Datasets." *IEEE Transactions on Circuits and Systems for Video Technology* (2021).

[9] Ren, Jingjing, et al. "Deep texture-aware features for camouflaged object detection." *IEEE Transactions on Circuits and Systems for Video Technology* (2021).

[10] Paterakis, Nikolaos G., et al. "Deep learning versus traditional machine learning methods for aggregated energy demand prediction." *2017 IEEE PES Innovative Smart Grid Technologies Conference Europe (ISGT-Europe)*. IEEE, 2017.

[11] Haq, Amin Ul, et al. "Comparative analysis of the classification performance of machine learning classifiers and deep neural network classifier for prediction of Parkinson disease." *2018 15th International computer conference on wavelet active media technology and information processing (ICCWAMTIP)*. IEEE, 2018.

[12] Howard, Andrew G., et al. "Mobilenets: Efficient convolutional neural networks for mobile vision applications." *arXiv preprint arXiv:1704.04861* (2017).

[13] Liu, Wei, et al. "Ssd: Single shot multibox detector." *European conference on computer vision*. Springer, Cham, 2016.

[14] Krizhevsky, Alex, Ilya Sutskever, and Geoffrey E. Hinton. "Imagenet classification with deep convolutional neural networks." *Advances in neural information processing systems* 25 (2012).

[15] Szegedy, Christian, et al. "Rethinking the inception architecture for computer vision." *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016

[16] Simonyan, Karen, and Andrew Zisserman. "Very deep convolutional networks for large-scale image recognition." *arXiv preprint arXiv:1409.1556* (2014).

[17] Kaiming, et al. "Deep residual learning for image recognition." *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016.

[18] Szegedy, Christian, et al. "Going deeper with convolutions." *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2015.

[19] Huang, Gao, et al. "Densely connected convolutional networks." *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2017.

[20] O'Shea, Keiron, and Ryan Nash. "An introduction to convolutional neural networks." *arXiv preprint arXiv:1511.08458* (2015).

[21] Soviany, Petru, and Radu Tudor Ionescu. "Optimizing the trade-off between single-stage and two-stage deep object detectors using image

difficulty prediction." *2018 20th International Symposium on Symbolic and Numeric Algorithms for Scientific Computing (SYNASC)*. IEEE, 2018.
[22] Redmon, Joseph, and Ali Farhadi. "Yolov3: An incremental improvement." *arXiv preprint arXiv:1804.02767* (2018).
[23] Kumar, Ashu, Amandeep Kaur, and Munish Kumar. "Face detection techniques: a review." *Artificial Intelligence Review* 52.2 (2019): 927-948.
[24] Zheng, Qiangqing, and Xiaolan Xie. "Traffic Sign Recognition Based on Learning Vector Quantization and Convolution Neural Network." *Proceedings of the 3rd International Conference on Intelligent Information Processing*. 2018.
[25] Fan, Deng-Ping, et al. "Camouflaged object detection." *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2020.
[26] Singh, Sujit K., Chitra A. Dhawale, and Sanjay Misra. "Survey of object detection methods in camouflaged image." *IERI Procedia* 4 (2013): 351-357.
[27] Zhou, Tao, et al. "RGB-D salient object detection: A survey." *Computational Visual Media* 7.1 (2021): 37-69.
[28] Gomez-Villa, Alexander, et al. "Convolutional neural networks can be deceived by visual illusions." *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2019.
[29] Gomez-Villa, Alexander, et al. "Color illusions also deceive CNNs for low-level vision tasks: Analysis and implications." *Vision Research* 176 (2020): 156-174.
[30] Lonnqvist, Ben, et al. "A comparative biology approach to DNN modeling of vision: A focus on differences, not similarities." *Journal of Vision* 21.10 (2021): 17-17.
[31] Zhang, Wei and Wang, Xiaogang and Tang, Xiaoou (2011). CUHK Face Sketch Database (CUFS), Retrieved June 17, 2022 from https://www.kaggle.com/datasets/arbazkhan971/cuhk-face-sketch-database-cufs
[32] Computational Intelligence and Photography Lab, Department of Computer Science, Yonsei University (2019) Real and Fake Face Detection, Retrieved June 17, 2022 from https://www.kaggle.com/datasets/ciplab/real-and-fake-face-detection
[33] Pushpak Bhoge (2021). Face Classification (real, drawing or illusion), Retrieved June 17, 2022 from https://www.kaggle.com/datasets/pushpakbhoge/fake-or-real
[34] K Scott Mader, ETH Zurich (2018). Food Images (Food-101), Retrieved June 17, 2022 from https://www.kaggle.com/datasets/kmader/food41
[35] Wong, Sebastien C., et al. "Understanding data augmentation for classification: when to warp?." *2016 international conference on digital image computing: techniques and applications (DICTA)*. IEEE, 2016.
[36] Lu, Lu, et al. "Dying relu and initialization: Theory and numerical examples." *arXiv preprint arXiv:1903.06733* (2019).
[37] Russakovsky, Olga, et al. "Imagenet large scale visual recognition challenge." *International journal of computer vision* 115.3 (2015): 211-252.
[38] Perez, Luis, and Jason Wang. "The effectiveness of data augmentation in image classification using deep learning." *arXiv preprint arXiv:1712.04621* (2017).
[39] Pasupa, Kitsuchart, Supawit Vatathanavaro, and Suchat Tungjitnob. "Convolutional neural networks based focal loss for class imbalance problem: a case study of canine red blood cells morphology classification." *Journal of Ambient Intelligence and Humanized Computing* (2020): 1-17.
[40] Lin, Zheng, et al. "Interactive image segmentation with first click attention." *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2020.
[41] https://en.wikipedia.org/wiki/Satyajit_Ray
[42] Rezatofighi, Hamid, et al. "Generalized intersection over union: A metric and a loss for bounding box regression." *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2019.