Article

Face Recognition Using Popular Deep Net Architectures: A Brief Comparative Study

Tony Gwyn [1,*], Kaushik Roy [1] and Mustafa Atay [2]

[1] Department of Computer Science, North Carolina A&T State University, Greensboro, NC 27411, USA; kroy@ncat.edu
[2] Department of Computer Science, Winston-Salem State University, Winston-Salem, NC 27110, USA; ataymu@wssu.edu
* Correspondence: tgwyn@aggies.ncat.edu

Abstract: In the realm of computer security, the username/password standard is becoming increasingly antiquated. Usage of the same username and password across various accounts can leave a user open to potential vulnerabilities. Authentication methods of the future need to maintain the ability to provide secure access without a reduction in speed. Facial recognition technologies are quickly becoming integral parts of user security, allowing for a secondary level of user authentication. Augmenting traditional username and password security with facial biometrics has already seen impressive results; however, studying these techniques is necessary to determine how effective these methods are within various parameters. A Convolutional Neural Network (CNN) is a powerful classification approach which is often used for image identification and verification. Quite recently, CNNs have shown great promise in the area of facial image recognition. The comparative study proposed in this paper offers an in-depth analysis of several state-of-the-art deep learning based-facial recognition technologies, to determine via accuracy and other metrics which of those are most effective. In our study, VGG-16 and VGG-19 showed the highest levels of image recognition accuracy, as well as F1-Score. The most favorable configurations of CNN should be documented as an effective way to potentially augment the current username/password standard by increasing the current method's security with additional facial biometrics.

Keywords: Convolutional Neural Networks; authentication; biometrics; face biometrics; facial recognition; classification methods

Citation: Gwyn, T.; Roy, K.; Atay, M. Face Recognition Using Popular Deep Net Architectures: A Brief Comparative Study. *Future Internet* **2021**, *13*, 164. https://doi.org/10.3390/fi13070164

Academic Editor: Steven Furnell

Received: 1 June 2021
Accepted: 22 June 2021
Published: 25 June 2021

1. Introduction

Biometrics are measurements of human characteristics that can be used for authentication purposes. These unique characteristics are nearly impossible to spoof, copy, or duplicate perfectly; this makes them an ideal candidate for increasing the security of user authentication. Facial biometrics in particular have shown great promise for authentication purposes, in part due to the way that user faces that can be accurately discerned and identified by systems [1,2].

In the past, the username/password standard was a sufficient level of security for most computer users; however, as time marches on, the methods of attackers and intruders have become more advanced. An intruder who is able to gain access to a user's computer, or who has a high level of knowledge about that user, could potentially be able to bypass the standard security of a computer system. As such, it has become necessary to augment or potentially replace the current username/password standard with a new system that uses facial biometrics to increase the level of security.

Several different systems have already been proposed to work with the username/password standard to increase the current state of computer security; unfortunately, while they do achieve various levels of success, they are not entirely without drawbacks. Although single sign-on WebIDs have shown effectiveness as a first line of defense against attacks [3], when deployed as a single system, their security as a standalone package is often lacking [4,5]. If an intruder is able to attain access to a user's unique certificate or their computer, they can quickly compromise a single sign-on WebID system if it does not have any additional security measures [6,7].

Conversely, WebIDs that have been enhanced with biometric authentication are harder to overcome, but this added security also increases the system's overall computational costs [8]. Additional schemas, including Eigenfaces [9,10] and Fisherfaces [11,12], work well at classifying images; however, their inability to handle the pace needed for real-time biometric authentication is an issue that can cause major problems when married to some authentication systems [13].

Various methods for facial recognition have also considered, including Artificial Neural Networks (ANNs) based on feed-forward classification. Several papers that use this type of classification indeed exist in the literature, including pattern recognition detection [14] and wavelet-based image classification [15]. In our research, it was determined that non-recurrent networks such as these, where information travels only in one direction, would be insufficient to the increasingly arduous process of facial image recognition [16].

The problem at hand is intricate and complex. As advances in technologies continue, and as attackers make use of such measures, the username/password standard is quickly becoming outdated. While several different attempts to increase the security of systems have been undertaken, those attempts have seen limited success. Further research into other methods, namely Convolutional Neural Networks (CNNs), should be investigated.

CNNs are powerful texture classification schemas which have been introduced to the realm of facial recognition with great success [17]. While the accuracy of CNNs has been thoroughly studied in a multitude of papers and journals, to our best knowledge, an extensive analysis to determine the best CNN for facial recognition has not yet been undertaken.

There are a multitude of CNN models available in the literature that have been used for facial recognition. These CNNs have various parameters that can be adjusted for the purpose of increasing accuracy with respect to facial recognition. In this study, a contribution to the area of facial recognition was made with our experiments into the efficacy of various types of CNNs.

We need to select the most favorable of these CNN models, and then provide an environment that puts these CNNs on a level playing field to compare their ability to properly discern faces. In addition, after testing these various CNNs, we need to establish a method for discerning which of these is best suited to the task of facial recognition. There are several different methods available for this task, including overall image accuracy as well as classification report metrics.

It was our goal for our proposed system to solve all the problems put forth by the previous questions. We undertook significant research into various CNN methods found in the literature prior to completing this work. Over 100 different methods were examined to determine their ability with regards to facial image recognition, which led to the selection of the eight current CNN models that we used in our study.

There currently exist many studies about deep learning models with respect to facial images, including for video surveillance [18], as well as more general surveys and analysis [19,20]. However, our work differed greatly from them in both the number of deep learning models that we tested as well as the scope of the CNNs that we selected for testing. Our main contributions in this study included the development of an image recognition system with face biometrics to extensively evaluate the accuracy of eight different CNN algorithms in a homogenous environment. The program tested these CNN models with respect to recognizing facial images using their default parameters to make sure the testing of these models occurred on as level a playing field as possible. We made use of the same dataset for all CNN methods, so that our results can be analyzed and compared among those methods.

We directly compared our models against each other using image recognition accuracy, as well as other accuracy measures such as a classification report. Section 2 describes our materials used in this paper; namely, the specific CNN variants, image dataset, and programming environment used. Section 3 defines the methodology of our experiments, including a further breakdown of the comparison of the eight different CNN models that were tested. Finally, Section 4 discusses the results that were obtained by the experiments, as well the conclusions gleaned from that work. It also outlines a pathway for potential future work with this topic.

While there are several different pre-trained networks already available, the experience gained from implementing and testing our own models in a very specific environment cannot be understated. Working with a complex set of CNN models allowed us to increase our own knowledge of neural networks with respect to facial image recognition. In addition, now that we have undertaken the challenge of establishing our own system for working with various CNNs, future implementation and testing of pre-trained networks can be undertaken with relative ease.

The CNNs that we determined to be best at image recognition accuracy were proposed for augmenting the current username/password standard, but only if the results we obtained met the rigorous standard required to provide that level of security. This study was directly influenced by previous work from the authors involving Local Binary Patterns (LBP) and facial biometrics [21].

2. Materials and Methods

This study was conducted via a Python algorithm customized for image recognition with various image classification schemas. The program made use of eight popular CNN models, along with a single image dataset for the testing of those models. While a plethora of different variations of these models and different parameters for these models exist in the literature, we made every effort to maintain their default parameters to ensure homogeneous testing. This allowed us to determine with a fair degree of certainty which of these CNN models performed the best with respect to image recognition accuracy.

2.1. CNN Variants Used

While there are easily dozens, if not hundreds, of image classification variations explored in the literature [17,22], the specific CNN variants in our proposed study have shown themselves to be particularly adept at the task of pattern recognition, especially in the realm of image classification [23,24]. The variants that are used in our study consist of eight Deep Nets: AlexNet, Xception, and then two versions each of the following: Inception, ResNet, and VGG. A comparison of each of these different CNN variants can be found in the following pages (Table 1), and a brief description of each of these CNN models can be found in Section 3.

Table 1. Comparison of different CNN Models used in our study.

CNN Model	Established	Total Layers	Conv Layers	Trainable Params	Unique Feature
AlexNet	2012	8	5	62,378,344	ReLU activation
Xception	2016	71	36	22,855,952	Depth-Seperable Convs
Inception v2	2014	48	22	55,813,192	Wider-Parellel Kernels
Inception v3	2014	48	22	23,817,352	Wider-Parellel Kernels
ResNet50	2015	50	48	25,583,592	Simpler Mapping
ResNet101	2015	101	99	44,601,832	Simpler Mapping
VGG16	2014	16	13	138,357,544	Fixed-size Kernels
VGG19	2014	19	16	143,667,240	Fixed-size Kernels

2.2. Image Dataset Used

For the testing of our CNN models, we decided to use a publicly available dataset that was specifically optimized for facial verification and recognition. To that end, we implemented the Labelled faces in the Wild (LFW) Image Dataset [25]. This dataset contains over 13,000 images from hundreds of different celebrities (Figure 1). The photographs are headshots that were collected from the web, labelled with the celebrity that the headshot belongs to. Unlike a typical facial database, where subjects have their images captured in a sanitized environment, often with a solid background, images in the LFW dataset come from real world environments; literally 'in the wild'.

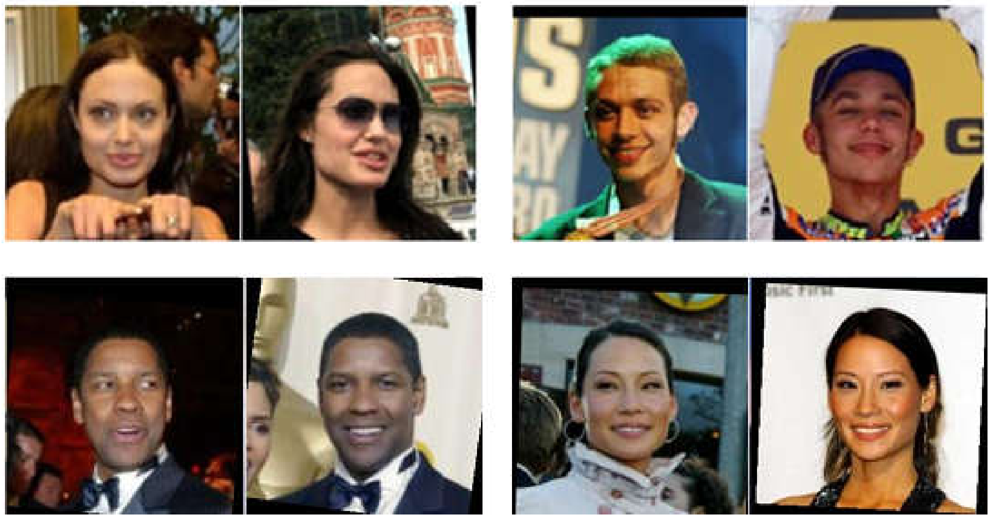
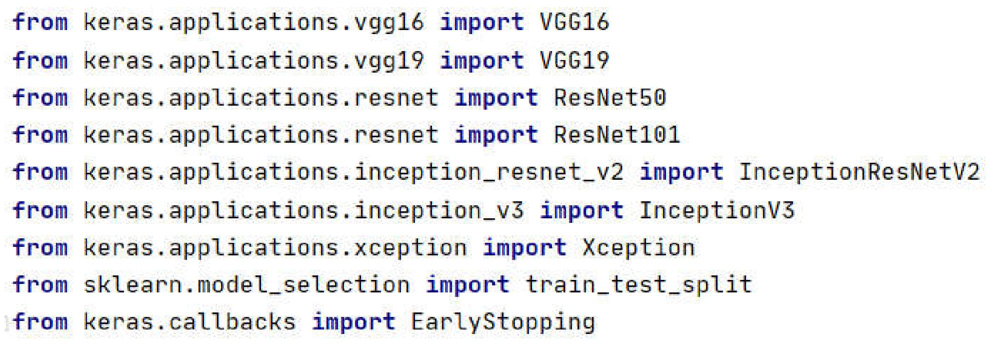


Figure 1. Example images of individuals from the Labelled Faces in the Wild (LFW) Image Dataset (http://vis-www.cs.umass.edu/lfw/, accessed on 23 June 2021).

Due to the specific training and testing split used with the images in this dataset, and because some of the celebrities had an insufficient number of images, our current testing environment did not use the entirely of the LFW Image Dataset. For the purposes of this particular study, we used 4788 celebrity images of 423 different celebrities, for an average of approximately 11 images per celebrity.

2.3. Python Environment

Our program made use of the Python programming language for the implementation and testing of our CNN models and dataset. Specifically, we used the PyCharm IDE, along with a handful of common Python packages that are used for image recognition purposes. The package that contained the versions of the CNN models we utilized came from the Keras open-source software library (Figure 2), which is specifically tuned for interfacing with artificial neural networks [26].



```
from keras.applications.vgg16 import VGG16
from keras.applications.vgg19 import VGG19
from keras.applications.resnet import ResNet50
from keras.applications.resnet import ResNet101
from keras.applications.inception_resnet_v2 import InceptionResNetV2
from keras.applications.inception_v3 import InceptionV3
from keras.applications.xception import Xception
from sklearn.model_selection import train_test_split
from keras.callbacks import EarlyStopping
```

Figure 2. Implementation of our CNN models using the Keras library.

2.4. Structure

After importing the Python packages needed for implementing image recognition, including TensorFlow, Pandas, and ImUtils, we subsequently imported the various CNN models we would utilize, as shown in Figure 2. As mentioned previously, we made use of the base CNN models wherever possible, so that adjusting most of the parameters in these models would not be necessary. The base parameters of these CNNs are already tuned for the purpose of object detection, which we made use of in facial recognition. This, in part, allowed for us to handle the implementation and execution of such a large variety of CNN models with relative ease.

We also introduced our dataset into the program at this time, and imported MatPlotLib for graph plotting functionality. Finally, we imported the packages needed for training and testing the model, as well as for handling the accuracy metrics, including the classification report.

Once the dataset was loaded into the program and the images were processed, the training of the models occurred. Depending on time and memory constraints, the models could be tested concurrently, or they could be split up to be tested one at a time. After the testing of the model(s) took place, the overall image recognition accuracy and classification reports were generated. It is at this time that we analyzed the data generated, and determined how each of the CNN models performed.

It should be noted here that the program was meant to be run a multitude of times, taking the average accuracy and other metrics generated by all of the runs. This was to gain a better picture of how each CNN model handled the data as a whole, and to discourage potential outliers that could indicate that a particular model performed better or worse than multiple tests would indicate.

3. Methodology

For the purpose of our testing, the models that performed poorly were documented, along with their accuracy and other metrics, while the CNN models that performed the best were also documented, and put forth as potential candidates for further scrutiny. These models bear further examination and analysis, as well as adjustment of their specific parameters towards the end goal of augmenting the current username/password standard with biometric authentication.

3.1. AlexNet

While Convolutional Neural Networks have always been the standard for object recognition, they do experience one problem: they are relatively hard to apply to higher resolution images. AlexNet is named after its designer, Alex Krizhevsky, who created the CNN architecture in conjunction with Ilya Sutskever and Geoffrey Hinton [27]. It was first brought to recognition when it competed in the ImageNet Large Scale Visual Recognition Challenge in 2012. A recreation of the AlexNet architecture can be seen in Figure 3 [27].

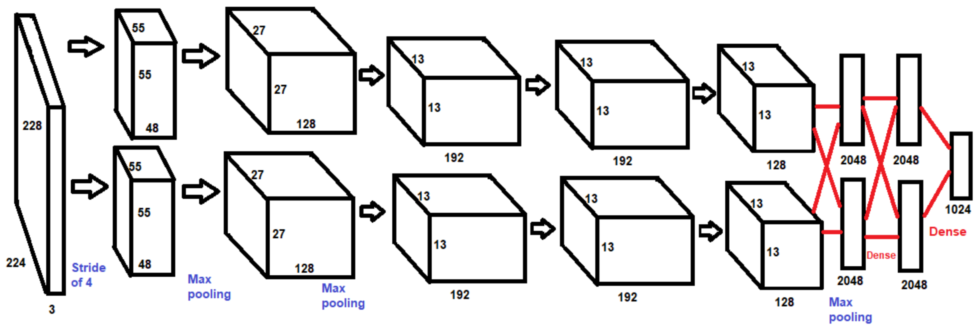


Figure 3. AlexNet architecture recreation, based on the original image found in [27].

AlexNet seeks to cut down on training times and establish optimizations for usage with Graphics Processing Units (GPUs), as well as increasing overall accuracy and performance. The model accomplishes this by making use of Rectified Linear Units (ReLU), as well as by incorporating multiple GPUs and establishing a process of overlapping pooling. The implementation of these novel methods allowed for AlexNet to see a decrease in training time and a reduction in errors, even with an increase in dataset size [28].

3.2. Xception

First proposed by Francois Chollet, this particular Convolutional Neural Network is adapted from the Inception CNN. The modules that one would typically find within Inception have been replaced with depthwise separable convolutions. The Xception model has nearly the exact amount of parameters as Inception v3, which is partially due to the fact that they share very similar architecture. A recreation of the Xception model architecture can be seen in Figure 4 [29].

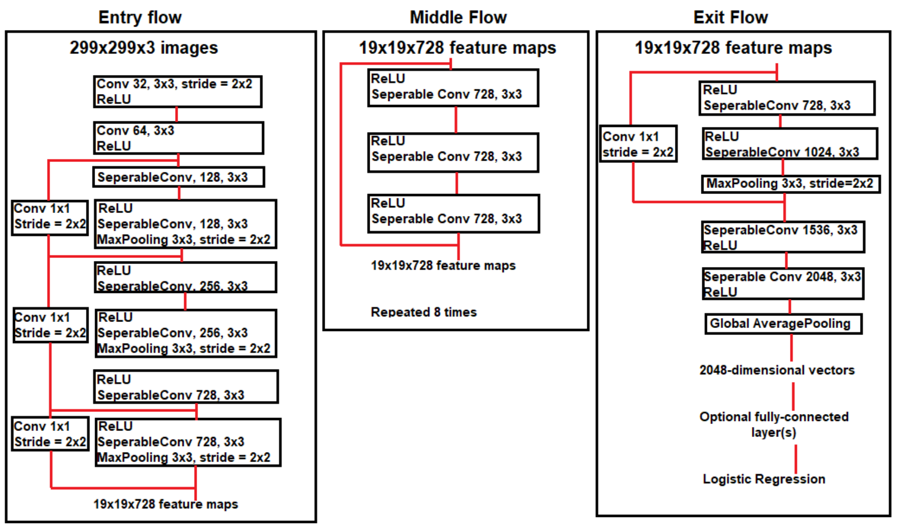

Entry flow
299x299x3 images
Conv 32, 3x3, stride = 2x2 ReLU
Conv 64, 3x3 ReLU
SeperableConv, 128, 3x3
Conv 1x1 Stride = 2x2
ReLU SeperableConv, 128, 3x3 MaxPooling 3x3, stride = 2x2
ReLU SeperableConv, 256, 3x3
Conv 1x1 Stride = 2x2
ReLU SeperableConv, 256, 3x3 MaxPooling 3x3, stride = 2x2
ReLU SeperableConv 728, 3x3
Conv 1x1 Stride = 2x2
ReLU SeperableConv 728, 3x3 MaxPooling 3x3, stride = 2x2
19x19x728 feature maps
Middle Flow
19x19x728 feature maps
ReLU Seperable Conv 728, 3x3
ReLU Seperable Conv 728, 3x3
ReLU Seperable Conv 728, 3x3
19x19x728 feature maps
Repeated 8 times
Exit Flow
19x19x728 feature maps
ReLU SeperableConv 728, 3x3
Conv 1x1 stride = 2x2
ReLU SeperableConv 1024, 3x3
MaxPooling 3x3, stride=2x2
SeperableConv 1536, 3x3 ReLU
Seperable Conv 2048, 3x3 ReLU
Global AveragePooling
2048-dimensional vectors
Optional fully-connected layer(s)
Logistic Regression

Figure 4. Xception recreation, based on the original image found in [29].

Xception could be considered an extreme version of Inception, because it takes the ideas put forth by Inception much further than any version of Inception thus far proposed [30]. Where Inception uses 1-by-1 convolutions for cross-channel correlations, and then captures spatial correlations with 3-by-3 or 5-by-5 convolutions, Xception instead performs 1-by-1 convolutions to every channel, then adding a 3-by-3 calculation to each of those outputs. This creates depthwise separable convolutions, which is what Xception uses to make its predictions.

3.3. VGG-16/VGG-19

Proposed by Karen Simonyan and Andrew Zisserman of the Oxford Robotics Institute, Visual Geometry Group (VGG) 16 and 19 are two well-established CNN models that work very well for the purpose of image classification and object localization [31]. In 2014, VGG-16 competed in the ImageNet Large Scale Visual Recognition Challenge (ILSVRC), where it attained first place in the object localization challenge, and second place in the image classification challenge. A recreation of the original VGG architecture can be seen in Table 2 [31].

Table 2. VGG recreation, based on the original image found in [31].

ConvNet Configuration					
A	**A-LRN**	**B**	**C**	**D**	**E**
11 weight layers	11 weight layers	13 weight layers	16 weight layers	16 weight layers	19 weight layers
input (224 × 224 RGB image)					
conv3-64	conv3-64 **LRN**	conv3-64 **conv3-64**	conv3-64 conv3-64	conv3-64 conv3-64	conv3-64 conv3-64
maxpool					
conv3-128	conv3-128	conv3-128 **conv3-128**	conv3-128 conv3-128	conv3-128 conv3-128	conv3-128 conv3-128
maxpool					
conv3-256 conv3-256	conv3-256 conv3-256	conv3-256 conv3-256	conv3-256 conv3-256 **conv1-256**	conv3-256 conv3-256 **conv3-256**	conv3-256 conv3-256 conv3-256 **conv3-256**
maxpool					
conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512 **conv1-512**	conv3-512 conv3-512 **conv3-512**	conv3-512 conv3-512 conv3-512 **conv3-512**
maxpool					
conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512 **conv1-512**	conv3-512 conv3-512 **conv3-512**	conv3-512 conv3-512 conv3-512 **conv3-512**
maxpool					
FC-4096					
FC-4096					
FC-1000					
soft-max					

VGG is able to use a relatively small architecture of 3-by-3 convolution features to attain impressive accuracy in image classification. The number associated with each VGG model is the number of total depth layers, the majority of those being convolutional layers [23]. The most widely used VGG models are VGG-16 and VGG-19, which are the two models that we chose for our study.

Despite being among the best CNN models at both object detection and image classification, VGG does have a few drawbacks which can make it challenging to use. Due to its robustness, VGG can be very slow to train; the initial VGG model was trained over a period of weeks on a state-of-the-art Nvidia GPU. Additionally, when VGG was utilized in the ILSVRC, the size of the weights used caused VGG to use a substantial amount of bandwidth and disk space.

3.4. ResNet50/ResNet101

As the task of image classification and recognition accuracy continues to become more complex, there is a need to create deeper and deeper neural networks to handle those challenges. Unfortunately, as additional layers are added, the difficulty in training those neural networks lead to a degradation in their accuracy. The specific architecture of ResNet was created to help solve this problem.

Introduced in 2015 by Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun, Residual Network (ResNet) is a type of CNN that is able to stack additional layers and attain increased performance and accuracy [32]. The layers that are added are able to learn more and more complex features, which in turn correlates to better system performance overall, as well as markedly increased image classification accuracy. This increase in layers needs to be balanced, as adding too many layers can cause an increase in error percentage as compared to a ResNet with fewer layers. An example of ResNet architecture can be found in Table 3 [32].

Table 3. ResNet architecture recreation, based on the original image found in [32].

Layer Name	Output Size	18-Layer	34-Layer	50-Layer	101-Layer	152-Layer
conv1	112 × 112	7 × 7, 64, stride 2				
conv2.×	56 × 56	3 × 3 max pool, stride 2				
		[3 × 3, 64] [3 × 3, 64] × 2	[3 × 3, 64] [3 × 3, 64] × 3	[1 × 1, 64] [3 × 3, 64] × 3 [1 × 1, 256]	[1 × 1, 64] [3 × 3, 64] × 3 [1 × 1, 256]	[1 × 1, 64] [3 × 3, 64] × 3 [1 × 1, 256]
conv3.×	28 × 28	[3 × 3, 128] [3 × 3, 128] × 2	[3 × 3, 128] [3 × 3, 128] × 4	[1 × 1, 128] [3 × 3, 128] × 4 [1 × 1, 512]	[1 × 1, 128] [3 × 3, 128] × 4 [1 × 1, 512]	[1 × 1, 128] [3 × 3, 128] × 8 [1 × 1, 512]
conv4.×	14 × 14	[3 × 3, 256] [3 × 3, 256] × 2	[3 × 3, 256] [3 × 3, 256] × 6	[1 × 1, 256] [3 × 3, 256] × 6 [1 × 1, 1024]	[1 × 1, 256] [3 × 3, 256] × 23 [1 × 1, 1024]	[1 × 1, 256] [3 × 3, 256] × 36 [1 × 1, 1024]
conv5.×	7 × 7	[3 × 3, 512] [3 × 3, 512] × 2	[3 × 3, 512] [3 × 3, 512] ×3	[1 × 1, 512] [3 × 3, 512] × 3 [1 × 1, 2048]	[1 × 1, 512] [3 × 3, 512] × 3 [1 × 1, 2048]	[1 × 1, 512] [3 × 3, 512] × 3 [1 × 1, 2048]
	1 × 1	average pool, 1000-d fc, softmax				
FLOPs		1.8×10^9	3.6×10^9	3.8×10^9	7.6×10^9	11.3×10^9

In our research, we determined that ResNet50 and ResNet101 were ideal candidates to be included in our study of CNNs. ResNet50 could be considered the standard version of ResNet, and has seen great success in the realm of image classification. ResNet101 has gone up against VGG-16 in several tests, and has seen impressive results, sometimes besting that particular CNN [33].

3.5. Inception v2/Inception v3

Prior to the creation of Inception, most of the popular CNNs were content to just stack convolution layers deeper and deeper on top of one another, in the hope of attaining better performance and higher accuracy. Inception worked in a different way; using a great deal of science, it was complex and heavily engineered. It was able to evolve quickly, leading to advancements and several different Inception versions that are still currently in use [34]. An example of the Inception architecture can be seen in Figure 5 [35].

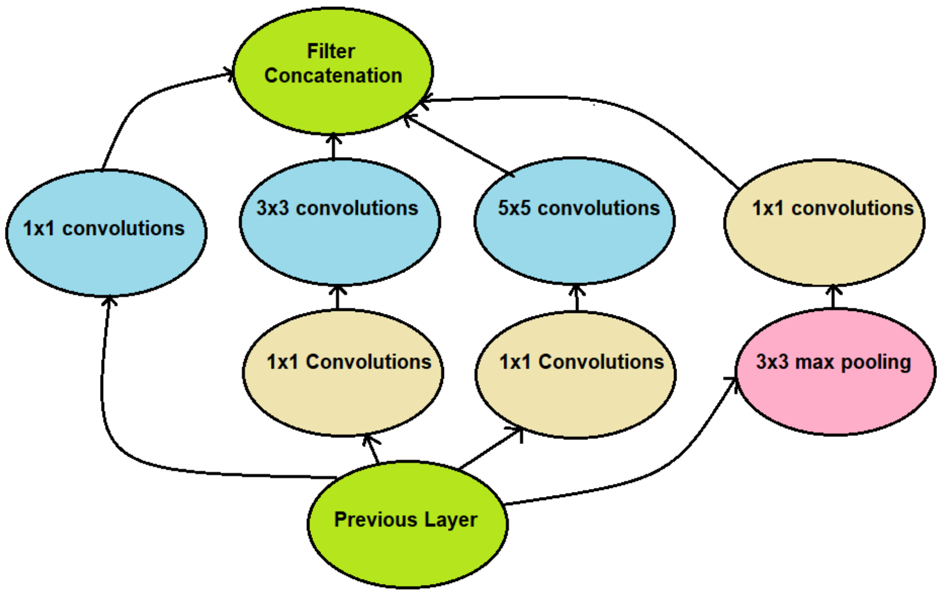


Figure 5. Recreated Inception architecture with dimension reductions, based on original image found in [35].

Inception hoped to solve the problem of the salient parts of images, which can vary in size depending on several factors. Because of this particular variation, it was extremely important for CNNs concerned with image classification to obtain the correct kernel size for convolution. A larger kernel size would be preferred for image information that is more global (the salient information is the entire image), whereas a smaller kernel would be useful where the image information is more local.

To solve this issue, the idea was to have filters that could have multiple sizes; instead of the network going deeper as is typical, it would instead go wider. Thus, Inception was conceived. In this particular paper, we made use of both Inception v2 and Inception v3 for use in our study. Inception v2 was proposed to reduce the potential bottleneck and loss of information from Inception v1, as well as to make the convolutions more efficient. For Inception v3, 7-by-7 convolutions were introduced, as well as some adjustments for auxiliary classifiers that were not significantly contributing to Inception v2. This leads to different versions of Inception that, while sharing some similarities, can have substantial differences in image classification [21].

4. Discussion

As noted previously, the purpose of this study was to determine which of the CNN methods currently being tested is the most effective with respect to accuracy and other metrics, including classification report. The most favorable method or methods would then be put forth as a candidate for further testing to potentially augment the current username and password authentication method with facial biometrics. A quick breakdown of our preliminary results can be found in the following figures.

4.1. Results

From the LFW dataset that was utilized, we had 4788 unique datapoints (images), and 423 labels (individuals). This led to a rather large imbalance in the dataset, which in turn caused a significant decrease in both recall and F1-Scores, the latter of which is reflected in Figure 6a. The accuracy of the dataset, when using an 80/20 testing/training split, can be seen in Figure 6b. A more complete breakdown of the classification report for all tested CNN models can be found in Table 4, which further shows the discrepancies between precision, recall, and F1-Score. We plan to incorporate more balanced datasets in the future to correct these issues.

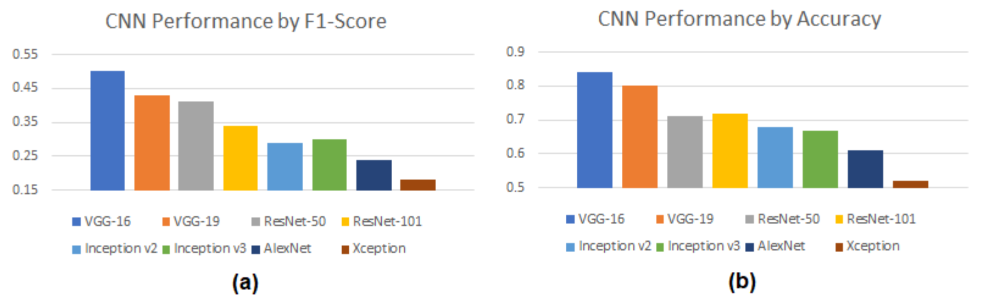


Figure 6. Results of CNN Image Classification testing by (**a**) F1-Score and (**b**) Accuracy, 80/20 Split.

Table 4. Further analysis of Classification Report for our tested CNNs, 80/20 Split.

CNN Model	Accuracy	Precision	Recall	F1-Score
AlexNet	0.61	0.66	0.15	0.24
Xception	0.52	0.57	0.11	0.18
Inception v2	0.68	0.7	0.18	0.29
Inception v3	0.67	0.75	0.19	0.3
ResNet50	0.71	0.76	0.28	0.41
ResNet101	0.72	0.75	0.22	0.34
VGG16	0.84	0.84	0.36	0.5
VGG19	0.8	0.8	0.29	0.43

As seen in Table 4, VGG-16 followed by VGG-19 performed best within the group with respect to both F1-Score and overall image classification accuracy. We believe this result is due to the novel way that VGG handles the convolutional layers, though it is a bit surprising that VGG-16 outperformed VGG-19, even though the latter utilized additional convolutional layers. The results we obtained for VGG-16 and VGG-19 are reasonable when compared to other comparative studies of CNN in the literature.

In addition to the typical 80 percent training, 20 percent testing split of the dataset, we have also included the results for both 70/30 and 60/40 training/testing splits. While we will focus most of the discussion on the typical 80/20 split, it is important to also note the results of the other splits that were tested, which can be found in Figures 7 and 8, respectively.

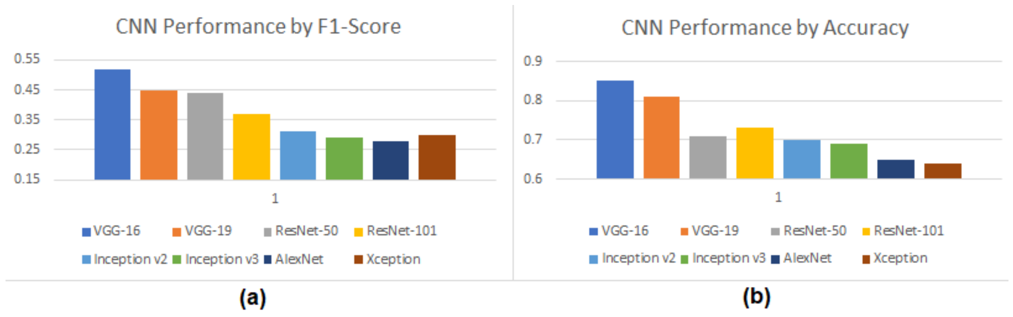


Figure 7. Results of CNN Image Classification testing by (**a**) F1-Score and (**b**) Accuracy, 70/30 Split.

In Figure 7, we see the results of changing the testing and training splits to 70% training images, and 30% testing images. Even with a smaller subset of images to train on, we saw a moderate increase in classification report metrics, as well as in overall image accuracy as opposed to the 80/20 training/testing split, as can be observed in Figure 7b. The largest gains made belong to AlexNet and Xception, which both saw increases in the measurable metrics. We postulate that, due to the particular way that both AlexNet and Xception work, the increase in testing images had a more marked effect on those than any of our other tested CNNs.

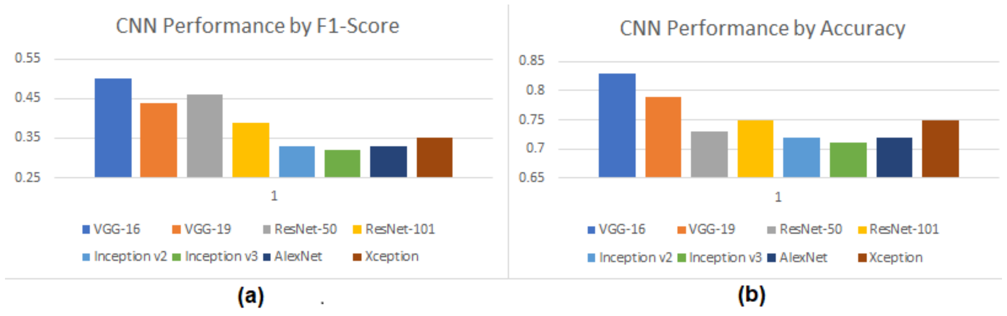


Figure 8. Results of CNN Image Classification testing by (**a**) F1-Score and (**b**) Accuracy, 60/40 Split.

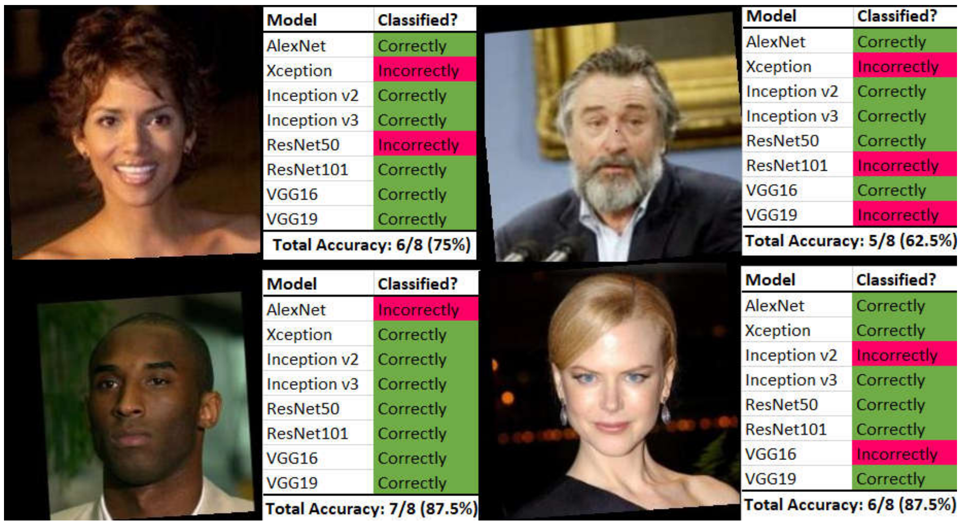
In Figure 8, we see the results of changing the testing and training splits to 60% training and 40% testing images. As with the previous increase in testing images, most CNNs saw an overall improvement in their classification report metrics and in their overall image accuracy. However, both VGG-16 and VGG-19 saw a slight reduction in the measured metrics. This could be due in part to the reduced amount of training images, which led to a plateau in the overall image classification accuracy for VGG-16 and VGG-19. For AlexNet and Xception, we again saw the best gains of any of our CNN models, which moved their performance from among the worst CNNs that we tested to somewhere in the middle of the pack. In contrast with the VGG models, in our experiments, AlexNet and Xception required less training on a reduced number of images, while still attaining impressive accuracy.

While both VGG-16 and VGG-19 showed impressive accuracy in all the tested configurations, they were not without their drawbacks. Although we are not currently factoring in processing time in our results, it should be noted that both VGG-16 and VGG-19 performed poorly with respect to those metrics, which can be viewed in detail in Table 5. Further testing will be conducted in the future to determine how much of an effect those two particular metrics may have on our results.

Table 5. Comparison of Computational Time between CNNs.

CNN Model	Trainable Params	80/20 Split	70/30 Split	60/40 Split
AlexNet	62,378,344	3 min 25 s	3 min 41 s	4 min 05 s
Xception	22,855,952	3 min 54 s	4 min 07 s	4 min 19 s
Inception v2	55,813,192	3 min 11 s	3 min 51 s	4 min 39 s
Inception v3	23,817,352	2 min 41 s	2 min 58 s	3 min 24 s
ResNet50	25,583,592	2 min 56 s	3 min 32 s	3 min 39 s
ResNet101	44,601,832	3 min 18 s	3 min 41 s	3 min 56 s
VGG16	138,357,544	7 min 33 s	8 min 39 s	10 min 24 s
VGG19	143,667,240	8 min 01 s	9 min 41 s	11 min 55 s

We have included here a brief example of correctly and incorrectly classified images with respect to some of the CNN architectures that were tested. While the models did reach a consensus on some of the images, several images were correctly classified by some of the models, and incorrectly classified by others. As can be viewed in Figure 9, there was sometimes a rather significant discrepancy, even between similar models (such as VGG-16 and VGG-19).



Model	Classified?
AlexNet	Correctly
Xception	Incorrectly
Inception v2	Correctly
Inception v3	Correctly
ResNet50	Incorrectly
ResNet101	Correctly
VGG16	Correctly
VGG19	Correctly
Total Accuracy: 6/8 (75%)	

Model	Classified?
AlexNet	Correctly
Xception	Incorrectly
Inception v2	Correctly
Inception v3	Correctly
ResNet50	Correctly
ResNet101	Incorrectly
VGG16	Correctly
VGG19	Incorrectly
Total Accuracy: 5/8 (62.5%)	

Model	Classified?
AlexNet	Incorrectly
Xception	Correctly
Inception v2	Correctly
Inception v3	Correctly
ResNet50	Correctly
ResNet101	Correctly
VGG16	Correctly
VGG19	Correctly
Total Accuracy: 7/8 (87.5%)	

Model	Classified?
AlexNet	Correctly
Xception	Correctly
Inception v2	Incorrectly
Inception v3	Correctly
ResNet50	Correctly
ResNet101	Correctly
VGG16	Incorrectly
VGG19	Correctly
Total Accuracy: 6/8 (87.5%)	

Figure 9. Example of correctly and incorrectly identified images by CNN model.

Moving clockwise from the top left image, we can see that both VGG models correctly identified the individual, yet there was a discrepancy between ResNet50 and ResNet101's conclusions. For the image at the top right, again ResNet50 and ResNet101 reached different conclusions, as did VGG16 and VGG19. The image at the bottom right found VGG16 and VGG19 again at odds, along with Inception v2 and Inception v3, which incorrectly and correctly identified the images, respectively. Finally, the image at the bottom left was correctly identified by seven of the eight models, with AlexNet being the only outlier.

It is interesting to see different conclusions being reached even among similar CNN models. In the case of ResNet50 and ResNet101, this discrepancy is likely due to the disparate number of trainable parameters between the two models; as for VGG16 and VGG19, the three extra convolutional layers in VGG19 led to that model finding different results from VGG16 in the top left and bottom left images. In addition, while both models of the Inception architecture found a consensus with three of the images in Figure 9, the disagreement caused by the bottom-left image could be due to the widening gap between their number of trainable parameters, or other minor differences between the two versions of Inception that we utilized.

4.2. Conclusion and Future Work

While there were clear winners in our experiments with respect to both accuracy and our other evaluation metrics, at this time none of the results rise to the level necessary to be considered as a candidate for facial biometric security enhancement. An accuracy level of greater than 80 percent, while reasonably high, does not meet the standard expected for augmentation of security protocols already in place. We need to continue to work towards increasing our level of accuracy; as such, we propose the following items for future work in this topic.

We plan to investigate and incorporate larger and more balanced datasets, and to test our image classification models against those datasets. Several such datasets exist in the literature, with not only a number of images far exceeding what we had available to us with the LFW dataset, but also with a more balanced number of images and labels. The proposed larger datasets will give our models more information, and should increase their ability to better discern the various labels that are provided.

While the eight CNN models that we tested had varying amounts of efficacy with respect to image classification accuracy, we need to investigate additional classification techniques for experimentation. Incorporating more CNN models, as well as more models of different image classification methods, will make our testing more robust and effective. This is necessary if we hope to achieve our goal of augmenting the current username/password standard with facial biometrics.

The CNN models utilized in this study were implemented and tested using only their default parameters. This effectively hamstrings these models, and limits their overall image classification accuracy. For future work in this topic, it will be necessary to incorporate a method of adjusting and examining the parameters available for each of our image classification methods, to ensure that we are achieving the highest image classification accuracy, as well as the best level of classification report possible. In the literature, many of these parameters have been tested and documented for several of the CNNs included in our study; however, there yet remain many different configurations that we should investigate further to best attain the highest image accuracy possible.

Finally, while image classification accuracy was paramount in this study, it would be unreasonable to ignore other evaluation factors. Security methods, especially those that work with real-time systems, need to be fast and agile; as such, processing time and memory allocation size play a role in the efficacy of these methods. In the future, we need to evaluate our image classification methods not only with accuracy metrics and classification reports, but also with respect to how much memory each model uses, as well as how much processing time it takes for that model to classify the images. It is only by incorporating these additional evaluation metrics that we will be able to truly put forth a candidate that can augment the current username/password standard.

Author Contributions: Conceptualization, T.G., K.R. and M.A.; methodology, T.G.; software, T.G.; validation, T.G.; formal analysis, T.G.; investigation, T.G. and K.R.; resources, T.G.; data curation, T.G. and K.R.; writing—original draft preparation, T.G.; writing—review and editing, K.R. and M.A.; visualization, T.G.; supervision, K.R.; project administration, K.R.; funding acquisition, K.R. All authors have read and agreed to the published version of the manuscript.

Funding: This research is supported by National Science Foundation (NSF). Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of NSF.

Institutional Review Board Statement: Ethical review and approval were waived for this study, due to the fact that a publicly available dataset was used (http://vis-www.cs.umass.edu/lfw/lfw.pdf, accessed on 24 June 2021).

Informed Consent Statement: Patient consent was waived due to the fact that a publicly available dataset was used (http://vis-www.cs.umass.edu/lfw/lfw.pdf, accessed on 24 June 2021).

Data Availability Statement: All data for this study is currently unavailable pending research maturity.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Ahonen, T.; Hadid, A.; Pietikäinen, M. Face Description with Local Binary Patterns: Application to Face Recognition. *IEEE Trans. Pattern Anal. Mach. Intell.* **2006**, *28*, 2037–2041, doi:10.1109/tpami.2006.244.
2. do Prado, K.S. Face Recognition: Understanding LBPH Algorithm. 2017. Available online: https://towardsdatascience.com/face-recognition-how-lbph-works-90ec258c3d6b (accessed on 10 November 2017).
3. Inkster, T.; Story, H.; Harbulot, B. WebID Authentication over TLS.; Last Retrieved July 2019. Available online: https://www.w3.org/2005/Incubator/webid/spec/tls/ (accessed on 23 June 2021).
4. Sabol, C.; Odd, W.; Esterline, A. Group Access Control using WebID. In Proceedings of the SoutheastCon 2016, Norfolk, VA, USA, 30 March–3 April 2016; pp. 1–2, doi:10.1109/secon.2016.7506672.
5. Sabol; C.; Nick; W.; Earl; M.; Shelton; J.; Esterline, A. The WebID Protocol Enhanced with Group Access, Biometrics, and Access Policies. In Proceedings of the MAICS 2016, Dayton, OH, USA, 22–23 April 2016; pp. 89–95.
6. Nick, W.; Shelton, J.; Sabol, C.; Esterline, A. Federated protocol for biometric authentication and access control. In Proceedings of the 2017 Computing Conference, London, UK, 18–20 July 2017; Institute of Electrical and Electronics Engineers (IEEE): Piscataway, NJ, USA, 2017; pp. 854–862.
7. Nick, W.; Murphy, A.; Sabol, C.; Esterline, A. A federated protocol for active authentication. In Proceedings of the SoutheastCon, Concord, NC, USA, 30 March–2 April 2017; pp. 1–2, doi:10.1109/secon.2017.7925377.
8. Pietikäinen, M.; Zhao, G. Two decades of local binary patterns. In *Advances in Independent Component Analysis and Learning Machines*; Academic Press: Cambridge, MA, USA, 2015; pp. 175–210.
9. Mainini, P.; Laube-Rosenpflanzer, A. Access Control in Linked Data Using WebID. 2016. Available online: http://arxiv.org/pdf/1611.03019.pdf (accessed on 9 November 2016).
10. Tomszuk, D.; Gebhardt, H.; Gaedke, M. WebID+ACO: A distributed Identification Mechanism for So-Cial Web. 2010. Available online: http://researchgate.net/publication/216818215_WebIDACO_A_distributed_identification_mechanism_for_social_web (accessed on 23 June 2021).
11. Sukhai, N.B. Access control & biometrics. In *InfoSecCD '04, Proceedings of the 1st Annual Conference on Information Security Curriculum Development, Kennesaw, Georgia, 8 October 2004*; Association for Computing Machinery (ACM): New York, NY, USA, 2004; pp. 124–127.
12. Kumar, K.K.; Pavani, M. LBP based biometrie identification using the periocular region. In Proceedings of the 2017 8th IEEE Annual Information Technology, Electronics and Mobile Communication Conference (IEMCON), Vancouver, BC, Canada, 3–5 October 2017; Institute of Electrical and Electronics Engineers (IEEE): Piscataway, NJ, USA, 2017; pp. 204–209.
13. Belhumeur, P.; Hespanha, J.; Kriegman, D. Eigenfaces vs. Fisherfaces: Recognition using class specific linear projection. *IEEE Trans. Pattern Anal. Mach. Intell.* **1997**, *19*, 711–720, doi:10.1109/34.598228.
14. Bonfitto, A.; Tonoli, A.; Feraco, S.; Zenerino, E.C.; Galluzzi, R. Pattern recognition neural classifier for fall detection in rock climbing. *Proc. Inst. Mech. Eng. Part P J. Sports Eng. Technol.* **2019**, *233*, 478–488, doi:10.1177/1754337119850927.
15. Kumar Singh, A.; Tiwari, S.; Shukla, V.P. Wavelet based Multi Class image classification using Neural Network. *Int. J. Comput. Appl.* 2012, *37*, 21–25, doi:10.5120/4597-6555.
16. Kasar, M.M.; Bhattacharyya, D.; Kim, T.-H. Face Recognition Using Neural Network: A Review. *Int. J. Secur. Appl.* **2016**, *10*, 81–100, doi:10.14257/ijsia.2016.10.3.08.
17. Han, S.-H.; Lee, K.-Y. Implemetation of image classification CNN using multi thread GPU. In Proceedings of the 2017 International SoC Design Conference (ISOCC), Seoul, Korea, 5–8 November 2017; Institute of Electrical and Electronics Engineers (IEEE): Piscataway, NJ, USA, 2017; pp. 296–297.
18. Bashbaghi, S.; Granger, E.; Sabourin, R.; Parchami, M. Deep Learning Architectures for Face Recognition in Video Surveillance. In *Deep Learning in Object Detection and Recognition*; Springer: Singapore, 2019; pp. 133–154, doi:10.10007/978-981-10-5152-4_6.
19. Chaudhuri, A. Deep Learning Models for Face Recognition: A Comparative Analysis. In *Deep Biometrics*; Springer Science and Business Media LLC: Berlin, Germany, 2020; pp. 99–140.
20. Shepley, A.J. Deep Learning for Face Recognition: A Critical Analysis. Available online: https://arxiv.org/ftp/arxiv/papers/1907/1907.12739.pdf (accessed on 12 July 2019).
21. Gwyn, T.; Atay, M.; Roy, K.; Esterline, A. Evaluation of Local Binary Pattern Algorithm for User Au-thentication with Face Biometric. In Proceedings of the 2020 19th IEEE International Conference on Machine Learning and Applications (ICMLA), Miami, FL, USA, 14–17 December 2020; pp. 1051–1058, doi:10.1109/IMCLA51294.2020.00170.
22. Pietikainen, M. Local Binary Patterns. 2010. Available online: http://www.scholarpedia.org/article/Local_Binary_Patterns (accessed on 23 June 2021).
23. Mahardi; Wang, I.-H.; Lee, K.-C.; Chang, S.-L. Images Classification of Dogs and Cats using Fine-Tuned VGG Models. In Proceedings of the 2020 IEEE Eurasia Conference on IOT, Communication and Engineering (ECICE), Yunlin, Taiwan, 23–25 October 2020; Institute of Electrical and Electronics Engineers (IEEE): Piscataway, NJ, USA, 2020; pp. 230–233.

24. Mahajan, A.; Chaudhary, S. Categorical Image Classification Based on Representational Deep Network (RESNET). In Proceedings of the 2019 3rd International conference on Electronics, Communication and Aerospace Technology (ICECA), Coimbatore, India, 12–14 June 2019; Institute of Electrical and Electronics Engineers (IEEE): Piscataway, NJ, USA, 2019; pp. 327–330.
25. Huang, G.; Ramesh, M.; Berg, T.; Learned-Miller, E. Labeled Faces in the Wild: A Database for Studying Face Recognition in Unconstrained Environments. October 2007. Available online: https://vis-www.cs.umass.edu/lfw/lfw.pdf (accessed on 23 June 2021).
26. Vani, A.; Raajan, R.N.; Winmalar, D.H.; Sudharsan, R. Using the Keras Model for Accurate and Rapid Gender Identification through Detection of Facial Features. In Proceedings of the 2020 Fourth International Conference on Computing Methodologies and Communication (ICCMC), Erode, India, 11–13 March 2020; Institute of Electrical and Electronics Engineers (IEEE): Piscataway, NJ, USA, 2020; pp. 572–574.
27. Krizhevsky, A.; Sutskever, I.; Hinton, G.E. Imagenet classification with deep convolutional neural networks. *Commun. ACM* **2017**, *60*, 84–90.
28. Lu, Y. Image Classification Algorithm Based on Improved AlexNet in Cloud Computing Environment. In Proceedings of the 2020 IEEE International Conference on Industrial Application of Artificial Intelligence (IAAI), Harbin, China, 25–27 December 2020; Institute of Electrical and Electronics Engineers (IEEE): Piscataway, NJ, USA, 2020; pp. 250–253.
29. Chollet, F. Xception: Deep Learning with Depthwise Separable Convolutions. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017; pp. 1251–1258.
30. Ayan, E.; Unver, H.M. Diagnosis of Pneumonia from Chest X-Ray Images Using Deep Learning. In Proceedings of the 2019 Scientific Meeting on Electrical-Electronics & Biomedical Engineering and Computer Science (EBBT), Istanbul, Turkey, 24–26 April 2019; Institute of Electrical and Electronics Engineers (IEEE): Piscataway, NJ, USA, 2019; pp. 1–5.
31. Simonyan, K.; Zisserman, A. Very deep convolutional networks for large-scale image recognition. *arXiv* **2014**, arXiv:1409.1556.
32. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep residual learning for image recognition. In Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 770–778.
33. Atliha, V.; Sesok, D. Comparison of VGG and ResNet used as Encoders for Image Captioning. In Proceedings of the 2020 IEEE Open Conference of Electrical, Electronic and Information Sciences (eStream), Vilnius, Lithuania, 30 April 2020; Institute of Electrical and Electronics Engineers (IEEE): Piscataway, NJ, USA, 2020; pp. 1–4.
34. Suresh, R.; Keshava, N. A Survey of Popular Image and Text analysis Techniques. In Proceedings of the 2019 4th International Conference on Computational Systems and Information Technology for Sustainable Solution (CSITSS), Bengaluru, India, 20–21 December 2019; Institute of Electrical and Electronics Engineers (IEEE): Piscataway, NJ, USA, 2019; Volume 4, pp. 1–8.
35. Szegedy, C.; Liu, W.; Jia, Y.; Sermanet, P.; Reed, S.; Anguelov, D.; Erhan, D.; Vanhoucke, V.; Rabinovich, A. Going Deeper with Convolutions. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Boston, MA, USA, 7–12 June 2015; pp. 1–9.