Real or fake: Detecting AI-Generated vs. Authentic Images

Differentiating between Artificial Intelligence Generated Images and Authentic Images: A Computer Vision Approach

Computational Vision

Hisham Unniyankal - 5049651

July 2023

1 Introduction

Artificial Intelligence (AI) has made significant advances in the generation of synthetic images, generating questions about the veracity of image data. Detecting AI-generated images has become essential for maintaining data integrity. In recent years, remarkable technological advancements have enabled the generation of images of such exceptional quality that it has become difficult for human observers to distinguish them from actual photographs.

This computer vision project seeks to address the growing difficulty of distinguishing between authentic photographs and those generated by artificial intelligence (AI). This project utilizes CIFAKE, a comprehensive dataset of 120,000 images that includes both real and artificial examples.

The primary objective of this project is to investigate various classification problem approaches in an effort to gain insight into the performance of various architectures. This project seeks to analyse the results of each classification method by experimenting with a variety of models, ranging from simple convolutional neural network (CNN) architectures to more complex ones such as VGG16, InceptionResnetV2, ALEXNET, and others. Through this exhaustive investigation, a deeper understanding of each architecture's performance characteristics can be attained.

2 Methods

This section will discuss the methodologies utilized in this project and the rationales underlying their selection. The dataset employed in this project is CIFAKE, which encompasses both authentic and artificially generated images. The dataset consisted of a total of 120,000 images, evenly divided between real and synthetic images, with 60,000 images in each category. The objective is to classify synthetic images and real images. Thus, the issue at hand can be framed as a binary classification problem, distinguishing between real and synthetic images.

2.1 Binary Image Classification

The Binary Image Classification algorithm is used to predict the class or label of a given input image. A trained model utilizes learned features extracted from input images and subsequently processes them to determine whether the image belongs to a real class or a synthetic class. Various Convolutional Neural Network (CNN) architectures were utilized in this study to accomplish the objective.

Convolutional Neural Networks(CNNs) are neural networks that has highly efficient performance on images. CNN architectures mainly built with three type of layers, convolution layers, pooling layers and fully connected layers. Convolution layers are the core building block of CNN which extracts features. It consists of trainable kernerls or filters extends fully into the depth of input. Convolutional leyers applies filters to the input image or previous feature maps to produce a new feature map. An activation function is perfomed on convolution layers to introduce non

linearity. This project mostly utilizes ReLu activation function.

Pooling layers are responsible for consolidating the features learned by CNNs. It reduces the representation's spatial dimensions to minimize the parameters and computation of network. Flattening output to a one-dimensional vector generates input vector to fully connected dense layers. Fully connected layers are responsible for recognizing and classifying images. Since the problem is a binary classification, final layer has a single neuron output with sigmoid as activation function.

Basic CNN Methods

The initial study was conducted utilizing just a few of the basic Convolutional Neural Network (CNN) architectures. Convolutional Neural Networks (CNNs) are well-suited for the task of image classification due to their ability to extract spatial hierarchies and local features from images. Initially, an architecture comprising of two convolutional layers, each followed by a max-pooling layer and a dropout layer, was used. The convolutional layers use 3x3 filters and ReLU activation functions. The output of the last convolutional layer is flattened and fed into a fully connected layer with 256 units and ReLU activation. The final layer is a single unit with sigmoid activation that outputs the probability of the binary class. Subsequently, an additional convolutional layer was incorporated into the architecture, and the experiment was repeated.

Dilated Network

Dilated CNN is a recent advancement in convolutional neural network which introduces one more hyper-parameter in to the convolutional layer. Dilation rate is a hyperparameter that determines the size of the gaps between the filter values in a dilated convolution operation. It is used to widen the kernel and increase the receptive field of the filter without increasing the number of parameters. Even though, dilated CNNs are not specifically designed for classification tasks, dilated convolution have generally improved performance in image classification tasks. It allows network to learn more complex features by capturing more context. Larger receptive field of dilated convolution enables the network to distinguish AI generated images by capturing more fine grained details and patterns.

Inception module

The inception modules uses multiple filters of different sizes to capture features of different sizes. Since the features that differentiate AI images from real images could be of different sizes, the idea of inception modules were utilized in this project. So that the model can learn features of different scales and resolution by using filters of different sizes. This makes the network more robust to variations in the size and shape of objects in the images.

The provided diagram in Figure 1, depicts the architectural structure of inception module. The project employs a rudimentary implementation of the inception module. The implementation in question employs three parallel convolutional filters with dimensions of 1x1, 3x3, and 5x5.

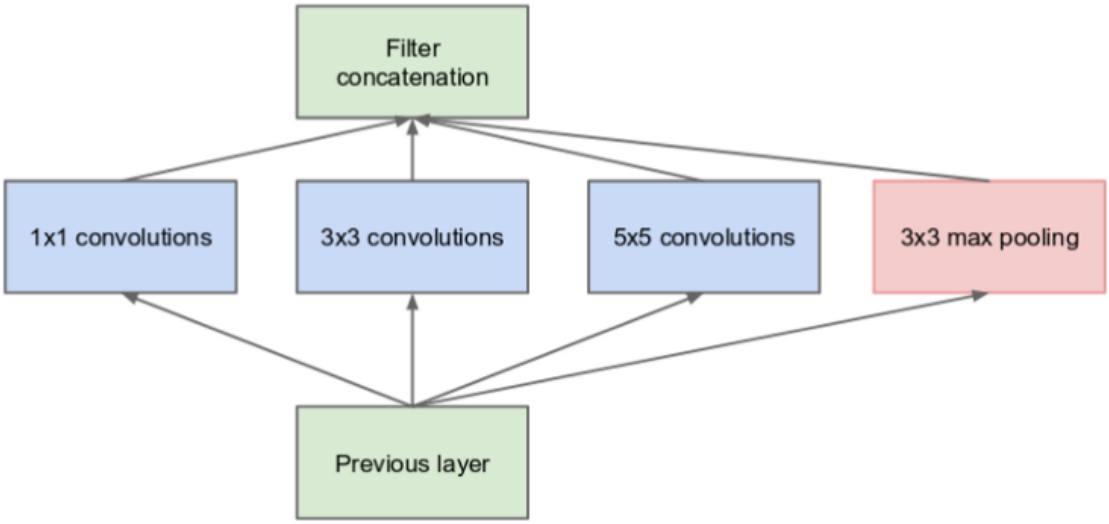

Figure 1: Architecture of single inception module

The inception module was introduced following the inclusion of two convolution layers, and subsequently, the project proceeded to conduct experiments involving multiple inception modules.

AlexNet Architecture

AlexNet is a convolutional neural network architecture proved to be highly accurate results in complex datasets. According to the paper 'ImageNet Classification with Deep Convolutional Neural Networks' by Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton, the architecture is good for classification. The experiments show that the AlexNet image classification algorithm improves the accuracy and stability of image classification and has good effectiveness and robustness. AlexNet is more complex than basic CNN architectures, but it has more layers and can learn more complex features.

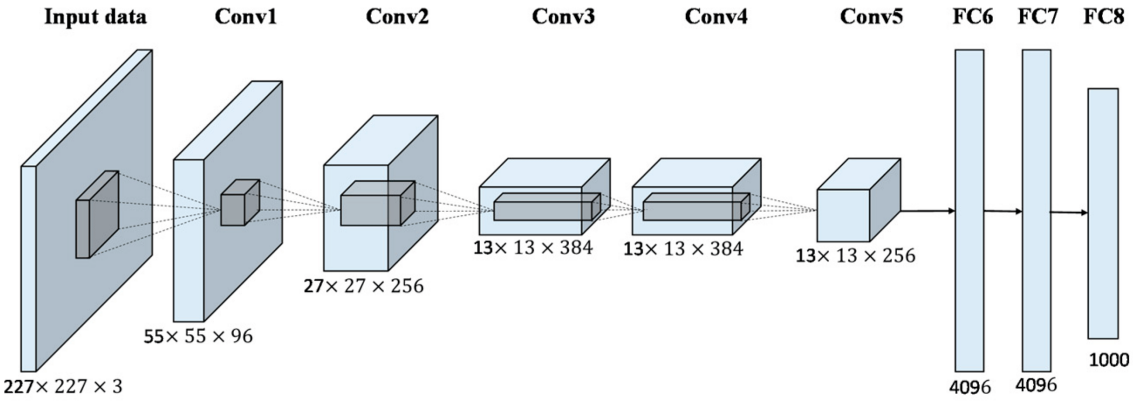

Figure 2: Alexnet CNN architecture

The architecture diagram of Alexnet is depicted in Figure 2. The architecture of AlexNet comprises five convolutional layers, succeeded by three fully connected layers. The initial convo-

lutional layer is comprised of 96 filters, each possessing dimensions of 11x11 and a stride value of 4. The second convolutional layer is composed of 256 filters, each with a dimension of 5x5 and a stride of 1. The fifth and sixth convolutional layers are equipped with 384 filters, each with a dimension of 3x3. The final convolutional layer consists of 256 filters with dimensions of 3x3. The initial two convolutional layers are subsequently accompanied by max pooling layers featuring a size of 3x3 and a stride of 2. Subsequent to the third, fourth, and fifth convolutional layers, there are max pooling layers with a 3x3 spatial extent and a stride of 1. The initial two fully connected layers comprise 4096 neurons, which are subsequently followed by a dropout layer. The final layer consists of a single neuron that is activated using the sigmoid function, resulting in a binary output.

InceptionResNetV2 architecture

The idea of using inception module led to another idea is to use of a pretrained model InceptionResNetV2. The architecture of InceptionResNetV2 combines the inception module and residual connections. The architecture is shown to achieve state-of-the-art performance on veriety of computer vision tasks. The idea of using this particular architecture is because its ability to capture high level features from images and its proven performance on similar kind of tasks.

The architecture of InceptionResNetV2 is highly complex and 164 layers deep. The complexity of the architecture made the training difficult. So the project opted a pretrained version of InceptionResNetV2 model. The model was pretrained on imagenet dataset. The last few layers of the model are fine-tuned to adapt it to the binary classification task.

VGG19

According to a journal published by Chandra Bhushana Rao Killi, Narayanan Balakrishnan, Chinta Someswara Rao earlier this year, VGG19 were provided accurate result on classifying between real and AI generated images. They proposed fine tuning VGG19 provided highly accurate results on deep fake images. The proposed work experimented on real and fake faces. The real faces were from Flickr and fake ones were generated using StyleGAN.

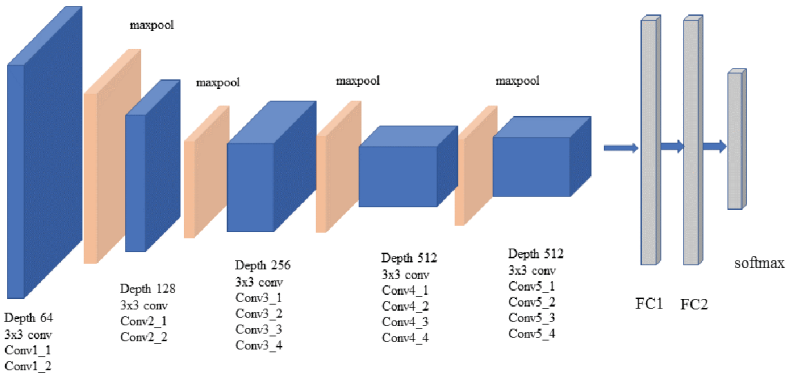

Figure 3: VGG19 CNN architecture

VGG19 is a DCNN architecture that has 19 layers and can classify images into 1000

object categories. In default, the network recieves an input shape of 224x224 and uses kernels of size 3x3 with a stride of 1 pixel and spatial padding to preserve the spatial resolution. The network consists of 16 convolutional layers, 5 max pooling layers, 3 fully connected layers and 1 softmax layer. The network has been pretrained on the ImageNet dataset, which contains over 14 million images of 1000 classes.

CNN as feature extractor

Another approach to the problem, proposed by the journal mentioned in section 2.6, is to use CNN as a feature extraction method. This way, the features extracted from the images by the CNN can be used as inputs for classical machine learning algorithms. In this project, different classifiers were experimented with: KNN, Logistic Regression and Gaussian Naive Bayes.

2.2 Evaluation metrics

This project evaluates all models with some of classification evaluation metrics. Evaluation metrics used in this project are accuracy_score, precision, recall and f1_score. This project also evaluates the ROC curve.

Accuracy score is one of the most fundamental and basic metric used to evaluate the model. Accuracy score measures that how many labels are correctly predicted.

$$\text{Accuracy} = \frac{\text{Number of Correct Predictions}}{\text{Total Number of Predictions}} \tag{1}$$

Precision : is another metric that computes the correctly predicted positives against all instances predicted as positive. It assess the model and quantifies the model's ability to avoid false positive predictions.

$$\text{Precision} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Positives}} \tag{2}$$

Recall : is a metric that measures the ability of a model to identify all relevant instances of class. This measures how many positive classes are correctly predicted.

$$\text{Recall} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Negatives}} \tag{3}$$

f1_score is calculated using both precision and recall to assess the model's performance. This metric is helpfull to evaluate when precision and recall have different trade-offs in importance.

$$\text{F1-Score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \tag{4}$$

ROC curve is a graphical representation of the trade-off between the true positive rate (TPR) and the false positive rate (FPR) for a binary classification model at different classification thresholds. True positive rate (TPR) is also known as recall and is defined as the proportion of actual positive cases that are correctly predicted by the model.

$$\text{TPR} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Negatives}} \tag{5}$$

False positive rate (FPR) is defined as the proportion of actual negative cases that are incorrectly identified by the model out of the total number of actual negative cases in the dataset.

$$\text{FPR} = \frac{\text{False Positives}}{\text{False Positives} + \text{True Negatives}} \tag{6}$$

2.3 Grad-CAM

Many of the existing deep learning methods are known to be black boxes, as they do not provide any reasoning for their classification decisions. Nevertheless, the project employs Gradient Class Activation Mapping (Grad-CAM), a technique that enables explainable AI. Grad-CAM interprets the gradients of the predicted class along with the CNN feature maps, which can be spatially localised with respect to the input image and generate a heatmap. Thus, Grad-CAM reveals the regions of the image that are most relevant for the classification outcome.

Grad-CAM employed in this project to visualize the features that causes the particular prediction. This project also used Grad-CAM to visualize some external images that are not from CIFAKE dataset.

3 Experimental Analysis

This section describes the data used in this project and provide a detailed explanation of the experiments carried out. This section also discuss the results of the experiments and provide a critical discussion of these results. The goal of this section is to provide a comprehensive overview of our experimental methodology and to present our findings in a clear and concise manner.

3.1 Dataset

The dataset employed in this project is the CIFAKE dataset, which comprises 120,000 images categorised into two classes: real and synthetic images. Both real and synthetic data exhibit a uniform distribution. The dataset utilised in this study comprises real data sourced from the CIFAR-10 dataset and the synthetic images are generated in a manner that is comparable to the CIFAR dataset, utilising the SDM model. All images in this dataset have dimensions of 32x32 pixels and are represented in the RGB colour space.

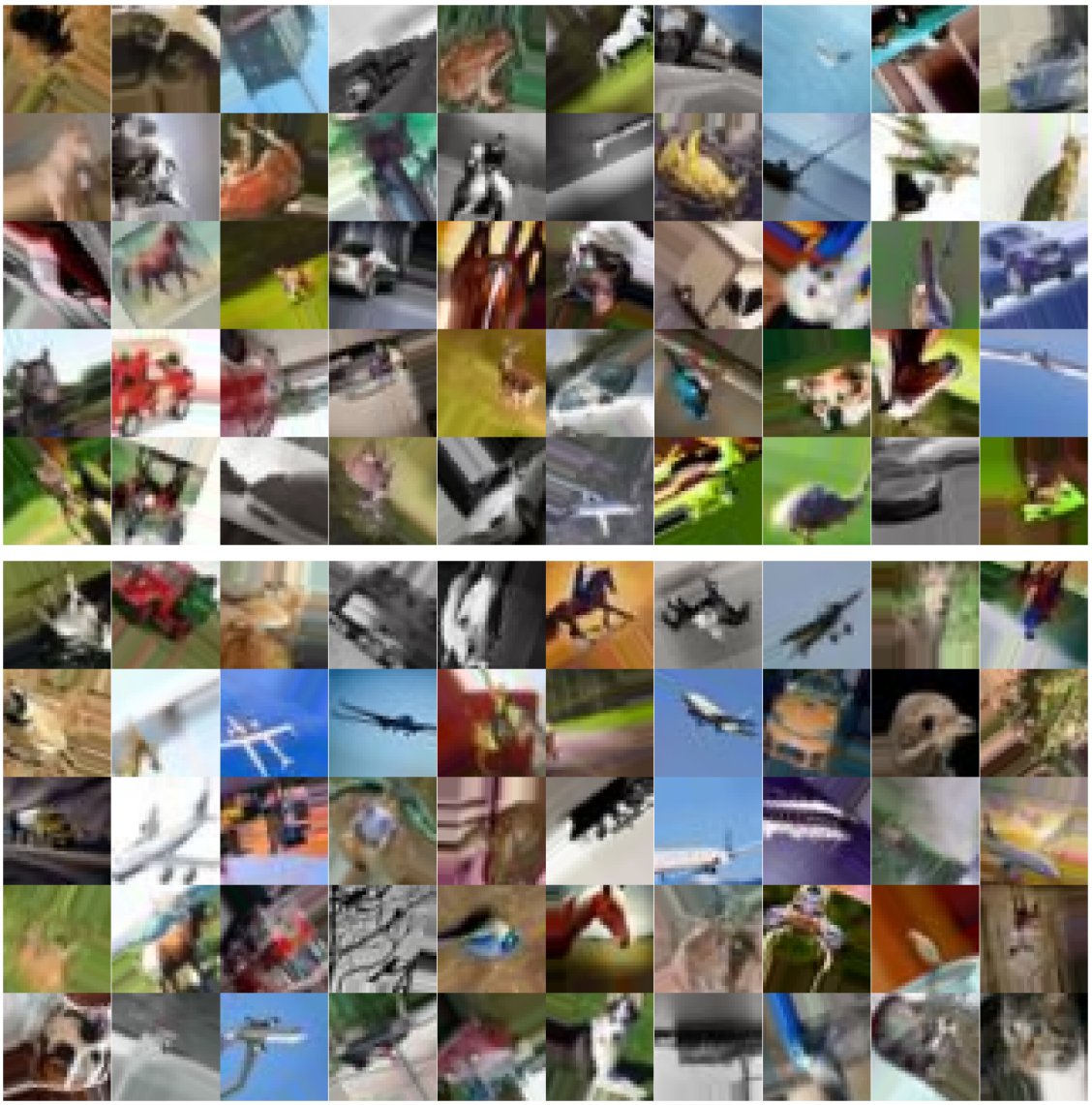

Figure 4: Real and fake images sampled from CIFAKE dataset.

Figure 4 displays a visual representation of both real and AI generated images that have been randomly selected from the CIFAKE dataset. The collection of images presented in the upper portion of the display corresponds to the real category, while the lower image collection pertains to the fake category. Distinguishing between synthetic and real images presents a significant challenge due to the close resemblance of synthetic images to their real counterparts. Nevertheless, it is possible that certain characteristics, such as defects and inaccuracies in the synthetic images, may play a crucial role in distinguishing the synthetic image from a real one.

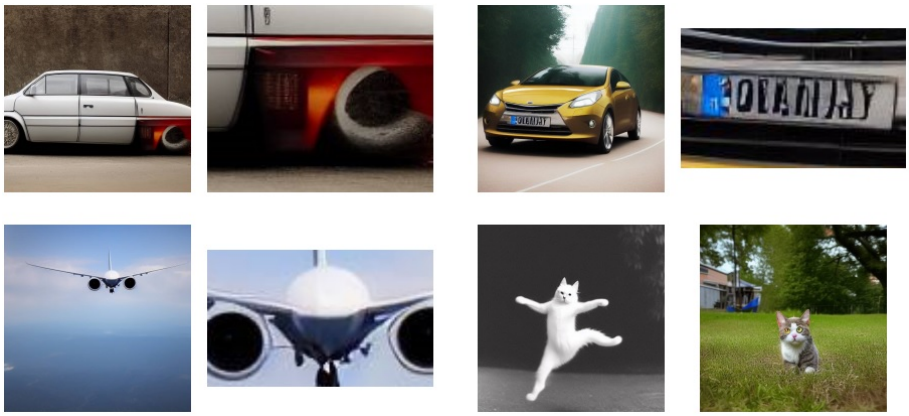

Figure 5: Examples of visual defects found within the synthetic image dataset(taken from journal mentioning CIFAKE).

Figure 5 depicts various potential anomalies that could potentially exist within a synthetic dataset. Upon careful examination of the initial depiction of the car, it becomes evident that there are certain discrepancies present within the image. Upon examination of the number plate depicted in the second image, it becomes evident that the text is illegible as a result of an error. In the scenario of a jet aircraft devoid of a cockpit window. Nevertheless, the number of such instances is limited, and in numerous images, the level of realism and complexity makes it challenging to differentiate between real and synthetic content.

3.2 Classification conducted and its results

This subsection will present the conducted experiment and the observed results. This problem primarily involves binary classification, wherein the objective is to accurately classify images as either real or AI-generated. Multiple architectures of Convolutional Neural Networks (CNNs) were evaluated and examined in order to compare and analyse the obtained outcomes.

Table 1 presents the recorded test loss and accuracy for each model architecture investigated in this research project. The performance of the AlexNet architecture was found to be superior when combined with the Nadam optimizer, with a learning rate of 0.001. The results indicated a test loss of 0.1655, which was the lowest observed, along with an accuracy rate of 93.43%. The CNN architectures employing two convolution layers exhibit a loss value of 0.4594, while those using three convolution layers demonstrate a higher loss value of 0.5571. Both achieved an accu-

Models	Loss	Accuacy
CNN: 2 Convolution Layers and 2 FC layers	0.4594	0.8064
CNN: 3 Convolution Layers and 2 FC layers	0.5571	0.7964
Dilated	0.2150	0.9165
Dilated with Nadam optimizer	0.2150	0.9298
Inception Module	0.4960	0.7930
Inception Module with Nadam optimizer	0.3747	0.8378
Alexnet	0.2014	0.9208
Alexnet with Nadam optimizer	0.1655	0.9343
Inception Resnet	0.2376	0.9020
VGG19	0.2376	0.8585

Table 1: Test accuracy and loss observed for experimented.

racy of 80.64% and 79.64% respectively. The inclusion of a dilation rate of 2 in the convolutional layers resulted in an enhancement of the model's performance, as evidenced by a decrease in loss to 0.2150 and an increase in accuracy to 92.98% when utilising the Nadam optimizer. An alternative proposal involved incorporating an inception module into the convolutional neural network (CNN) architecture, resulting in relatively inferior performance when compared to other architectures. In order to conduct additional experimentation, pretrained models, namely Inception Resnet and VGG 19, were employed. The results indicated that Inception Resnet achieved an accuracy of 90.20%, while VGG19 achieved an accuracy of 85.85%.

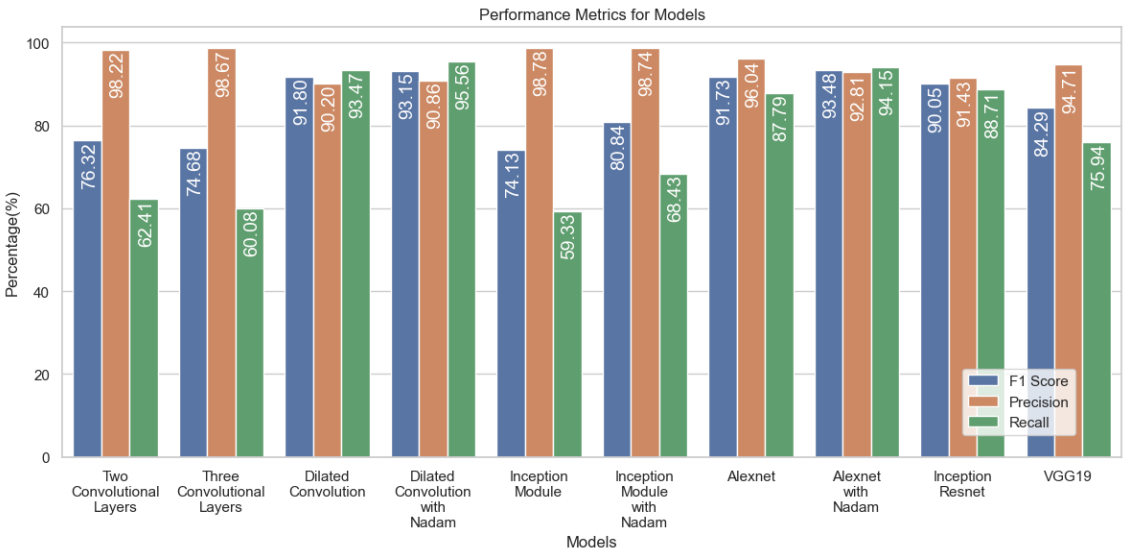

Figure 6: Comparison of Performance Metrics (Accuracy, F1 Score, Precision, Recall) across experiments: Bar Chart

Relying solely on the accuracy score may not be a enough for gaining a comprehensive understanding of a model's performance. Upon examination of Figure 6, it is evident that the

bar chart illustrates that all of the models exhibited a greater level of precision. The model incorporating dilated convolution layers has demonstrated a minimum observed precision of 90%. The precision scores of architectures comprising 2 convolution layers, 3 convolution layers, inception modules, and inception modules with the nadam optimizer were observed to exceed 98%. However, these models exhibited a relatively low recall rate, ranging from 60% to 70%. The accuracy score of the AlexNet architecture with the Adam optimizer was observed to be 92.08%, which is relatively high compared to some of other models. Nevertheless, the precision of the aforementioned model is 96.04%, while the recall stands at 87.79%. These values indicate a minor discrepancy in the prediction of classes. In order to enhance comprehension of performance evaluation, the F1 score is used. This metric effectively amalgamates precision and recall, enabling the selection of an optimal model that strikes a trade off between these two measures. The utilisation of the AlexNet architecture, incorporating dilated convolution and the nadam optimizer, resulted in improved f1 scores of 93.46% and 93.15%.

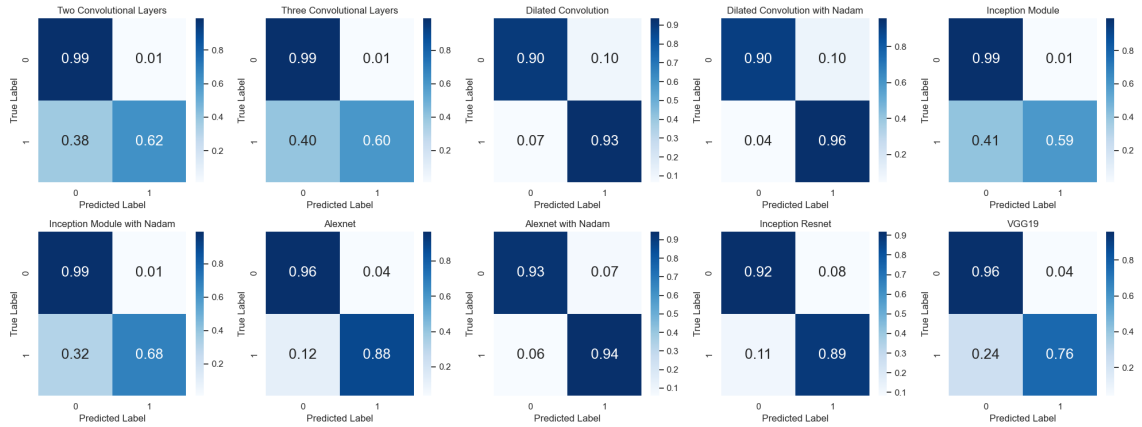

Figure 7: Confusion matrix for CNN architectures

Figure 7 displays the confusion matrix and Figure 8 shows ROC curve for each of the model architectures. Both plots demonstrate that the AlexNet and dilated convolution with Nadam optimizer yield a more balanced prediction compared to other architectures that were tested. Both models exhibit an AUC (Area Under the Curve) of 0.93, while basic CNN architectures and architectures with inception modules demonstrate lower values of 0.79 and 0.80, respectively.

Hence the architecture with dilated convolution layers provides a better prediction, the project then used the model as a feature extractor to investigate how the machine learning algorithms works for this task. The remaining layers were pruned, resulting in a model that is now employed for feature extraction. Figure 9 illustrates the performance of the machine learning algorithms when applied to features derived from convolutional neural networks (CNN). This study employed the K-nearest neighbours (KNN) classifier, logistic regression, and Gaussian naive Bayes model to classify between real and generated images.

The K-nearest neighbours (KNN) classifier achieved an accuracy of 92.84% and an F1 score of 92.73%. The logistic regressor exhibited a slight improvement, achieving an accuracy of

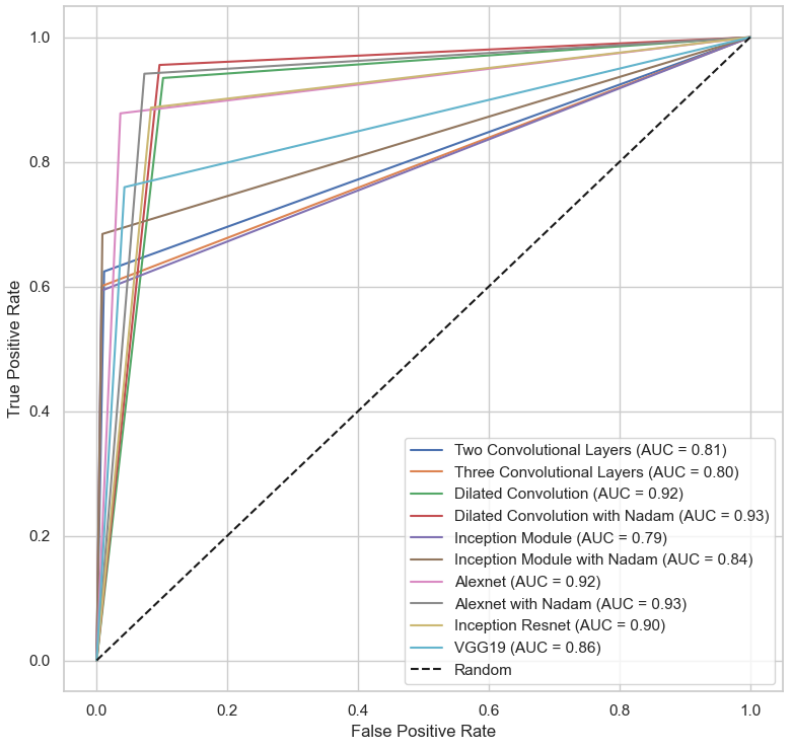

Figure 8: ROC curve

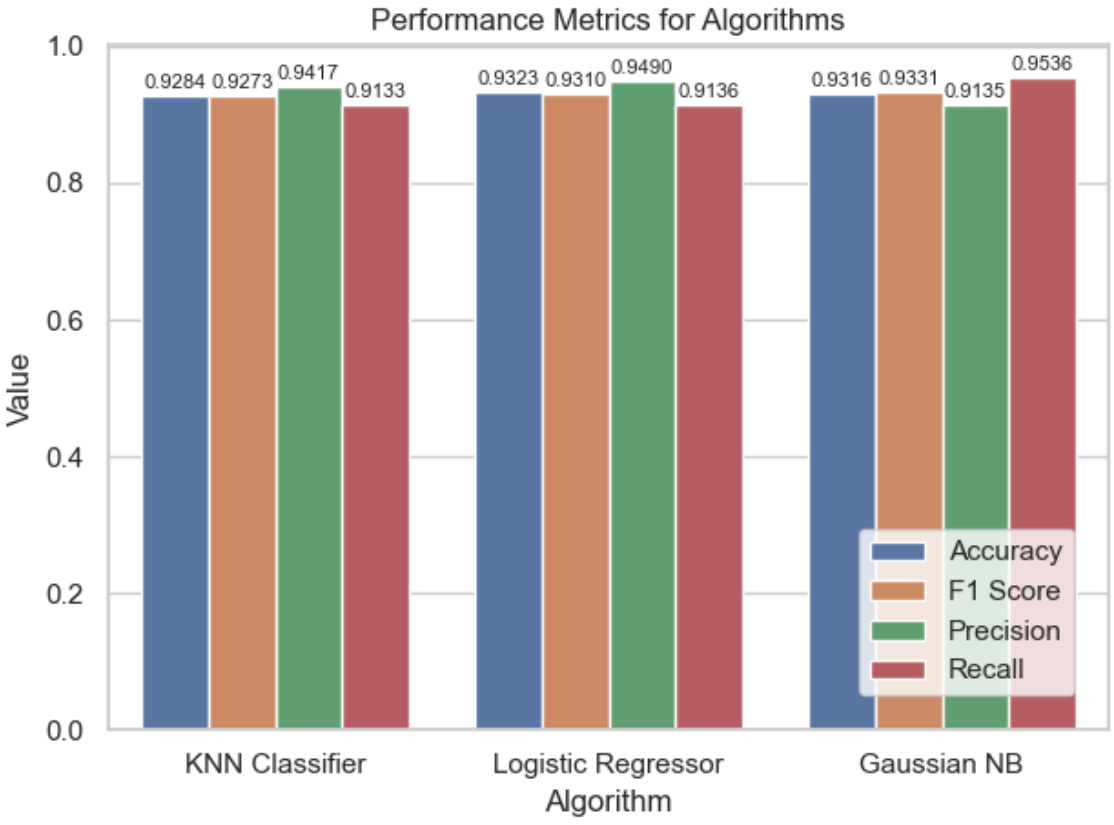

Figure 9: Barchart for machine learning algorithms

93.23% and a f1 score of 93.10%. Additionally, the Gaussian NB model demonstrated comparable performance to the logistic regressor, achieving an accuracy of 93.16% and a f1 score of 93.31%. When comparing the performance of machine learning models to the dilated convolution with dense layer using sigmoid for classification, it was observed that the Logistic regression and Gaussian NB produced slightly superior results.

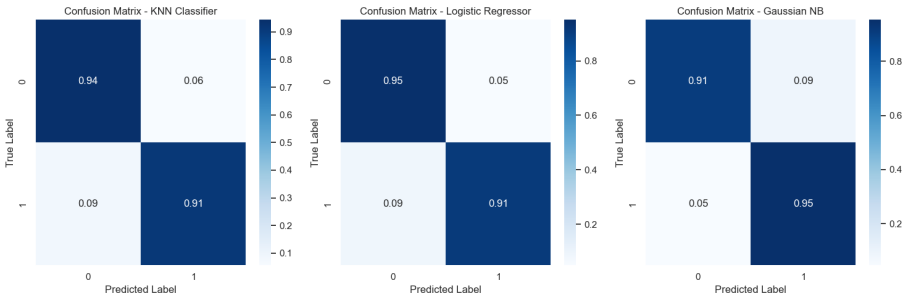

Figure 10: Confusion matrix for machine learning algorithms

The interpretation of predictions utilising gradient class activation mapping is depicted in Figure 11. The presence of bright pixels in a heat map signifies a higher degree of contribution towards the prediction. The heatmap generated through the utilisation of Grad CAM exhibits a notable disparity in the distribution of features. In the lowermost figure, the examination of heatmaps reveals a high density of features in relation to real class prediction. This indicates that a substantial number of features from real images are contributing to the prediction. In the context of predicting fake classes, the majority of these cases exhibit limited or negligible feature contributions.

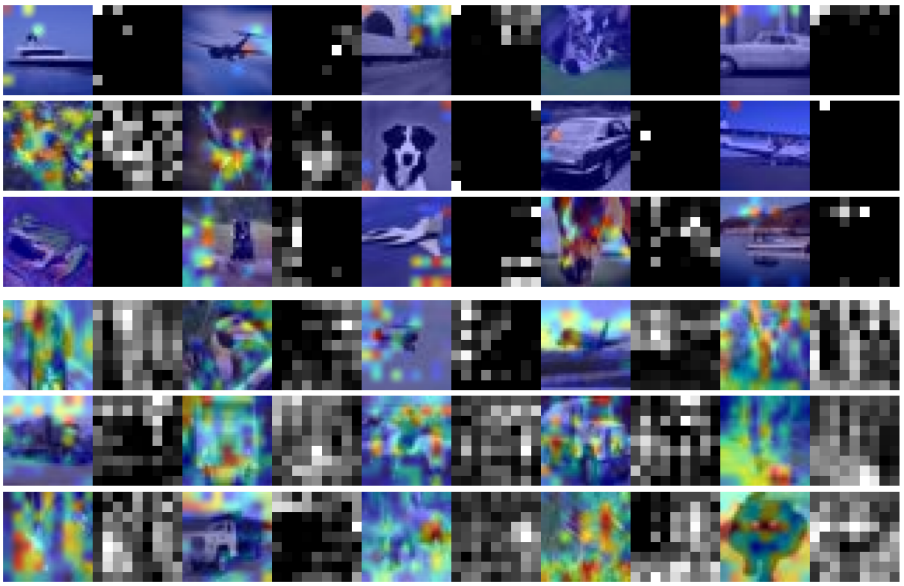

Figure 11: Gradient class activation maps (Grad-CAM) overlays and raw heatmaps for prediction interpretation. Top examples show real images and bottom examples show AI-generated images. Brighter pixels represent features contributing to the output class label.

4 Conclusion and Future Works

The objective of this project was to design and develop a idea capable of distinguishing between AI-generated images and real images. In order to accomplish this objective, the project conducts experiments on different Convolutional Neural Network (CNN) architectures. The study employed Gradient Class Activation Mapping (Grad-CAM) as a method for interpreting image predictions. Several models were tested in the project, and few of them yielded a notable outcomes with great accuracies upto 93.48%.

The advent of Artificial Intelligence and the development of synthetic image generation have posed a significant challenge to the veracity of images. Comprehending it poses a significant challenge within the realm of human perception. This project employs computer vision techniques to comprehend images generated by artificial intelligence.

Potential future research could entail conducting a comprehensive exploration of more sophisticated convolutional neural network (CNN) architectures, and subsequently conducting a comparative analysis of their performance in relation to the currently existing ones. For example, study ensemble methods as a means of integrating the advantages of multiple convolutional neural network (CNN) architectures in order to enhance accuracy and robustness in image classification tasks. An additional suggestion entails the investigation of a broader range of authentic and artificially generated images from different sources, thereby augmenting the complexity of the dataset.