Assignment 4: Fourth Research/Programming Assignment

Kathryn LiPetri

MSDS 458: DL Section 56, Winter 2023

March 12, 2023

Abstract

Transfer learning was used to conduct experiments to develop and fine-tune a model that can identify geographic locations that are at risk of wildfire, or no risk of wildfire occurrence based on color satellite images of the locations. Experiments were conducted using Python programming framework with pre-trained model VGG16 in Keras imported from Tensorflow 2.11.0. Data from Canada's website for forest fire mapping was used and included color satellite images to train, validate, and test the model. Variation of model architecture and hyperparameters were also used to optimize the model for identification of areas at high risk of wildfire. The resulting model can classify satellite images of areas with high risk of wildfire and areas with no risk of wildfire with 93% accuracy.

Introduction

Each year, wildfires occur in specific parts of the world, whether caused by human activity or nature, i.e., lightning. Wildfires can burn for a very long time and cause a lot of destruction and pollution, not to mention injuries and deaths. A government in an area that experiences frequent wildfires, such as California, may want to be able to identify locations that are at high risk of wildfire. This way preventive measures can be implemented in a targeted way to try to decrease the risk of wildfire, and people nearby can be warned to be more careful with open flames, setting off fireworks, etc.

The goal of this project is to develop a model that will be able to identify if a location is at risk of wildfire occurrences using computer vision and satellite images of specific geographic locations as data to be classified.

For this assignment, a neural network will be created using transfer learning with the pre-trained VGG16 network to classify satellite images of areas at high risk and low risk of wildfires

with a target of 90% accuracy or more. Data used to train and test the model came from Canada's website for forest fire mapping and were extracted from satellite images and put into a format suitable for deep learning and modeling on Kaggle (Aaba 2023). The data available contains satellite images in the classes of "Wildfire" with 22,710 images and "No Wildfire" with 20,140 images. Each image is color and 350x350 pixels. Data were split into training, validation, and test datasets of 30,025 training images, 6,300 validation images, and 6,300 test images, however only 10% of available data (3,025 training images, 630 validation images, and 630 test images) was used for this project because of limitations on computing power. Experiments were conducted using Python programming framework with pre-trained model VGG16 in Keras with variations made to model architecture and hyperparameters.

Literature Review

The technique used for image classification is known as Computer Vision. Computer Vision is a field of Artificial Intelligence that trains computers to see and process images as humans would. Pattern recognition is a large part of Computer Vision, so a computer must be trained on thousands or millions of labeled images for it to understand and begin to be able to decipher visual data. For a computer to understand color images, such as the satellite image data being used for this assignment, the images are converted to pixels that each have three values corresponding to red, green, and blue with each color having a value in the range from 0 to 255 (Mihajlovic 2019).

In the 1950s and 1960s, through research it was determined that human vision is hierarchical. This means that neurons first detect simple features, like edges, then more complex features like shapes and eventually complex visual representations. This knowledge of human vision has been the foundation of recreating human neurological structures in digital form for

computer vision. Initial experiments in computer vision in the 1960s involved attempting to connect a camera to a computer and having the computer describe what it saw. This experiment did not work well, and faith in artificial intelligence was lost for a few decades. Once computer scientists had access to the internet with larger amounts of data and less expensive hardware, neural networks and algorithms for computer vision were improved. Now, neural networks have revolutionized the field of artificial intelligence, and computers are able to perform computer vision tasks with very low error rates (Motion Metrics 2019).

For this assignment, Computer Vision is being used for image classification. Image classification is a fundamental problem in the field of Computer Vision and has many different applications. Image classification can be performed using supervised or unsupervised techniques. Unsupervised techniques are fully automated and do not utilize training data. An example of this is K-means clustering, which is an algorithm that groups objects into groups based on their characteristics. For this assignment, supervised learning will be used to classify satellite images. This means that a model will be trained using previously classified reference samples before it is then used to classify new, unknown data. Deep learning is the most popular machine learning technique for image classification tasks. The CNN (Convolutional Neural Network) is a deep neural network used for computer vision tasks because its multilayer neural network is inspired by the mechanism of a human optical and neural system (Boesch 2023).

In addition, instead of building a neural network from the ground up, transfer learning will be used to model the “Wildfire” or “No Wildfire” classifier. Transfer learning is a technique in machine learning that applies knowledge already gained when solving a problem to the solving of the next problem. This involves using a pre-trained model for the given problem. The benefits of transfer learning are that it takes less time to train the model since it has already been

trained on the previous problem and it can be used for smaller datasets since the model has already been trained on a larger dataset. There are many different pre-trained models available in the Tensorflow Keras module to be used for transfer learning purposes. The pre-trained model chosen for this project is the VGG16 model. The VGG16 model is a convolutional neural network (CNN) that was proposed in 2014 at the ImageNet Large-Scale Visual Recognition Challenge by Karen Simonyan and Andrew Zisserman of the Visual Geometry Group Lab of Oxford University. The model was trained on the ImageNet dataset which contains 14 million images belonging to 1000 classes. The VGG16 model can achieve 92.7% test accuracy on the ImageNet dataset (VGG-16 2023).

Methods

For this assignment, transfer learning was used to fine tune a pre-trained CNN model, VGG16, to classify geographic satellite images as "Wildfire" (risk of wildfire occurrence) or "No Wildfire" (no risk of wildfire occurrence). CNNs are neural networks that contain at least a convolutional layer as the first layer and a fully connected layer as the last layer. Unlike dense neural networks, all nodes are not connected in a CNN because convolutional layers extract only relevant features to pass to further layers in the network. This allows CNNs to have a reduced number of parameters compared to dense neural networks, therefore less training data is needed for a CNN to get the same results as a dense neural network with a larger training set. The final output layer is fully connected for image classification to occur based on features that were extracted in the previous layers. Often, a pooling layer will also be added after the convolutional layer for the purpose of reducing the size of an image. This is done by pooling inputs from the previous layer into one output, either the largest input value in the pool for "max pooling" or the average of all inputs in the pool for "average pooling". Before the output layer, features from the

pooling layer are flattened into a one-dimensional array so that each feature can connect with the final layer.

The VGG16 model has a CNN architecture containing 16 layers that have weights (trainable parameters). The model has 13 convolutional layers, 5 max pooling layers, and 3 dense layers, totaling 21 layers. See Figure 1 for a simple depiction of the VGG16 model architecture:

Figure 1. VGG16 Architecture

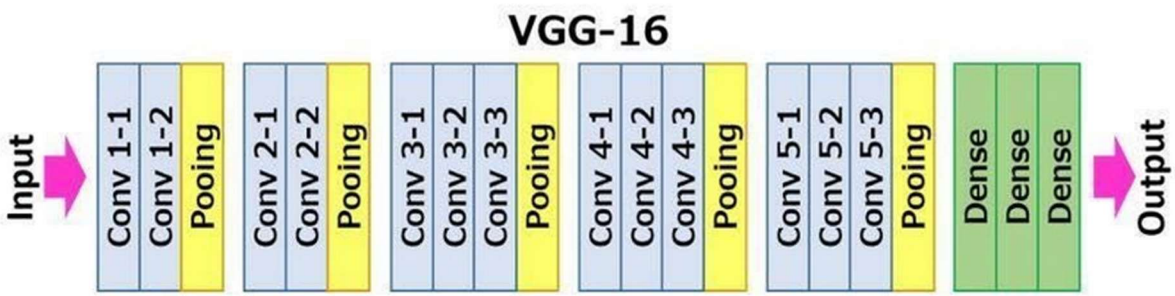


Figure 1 G, Rohini 2021

The model has been trained on the ImageNet dataset of 14 million images from 1000 classes with images of size (224, 224, 3). VGG16's convolutional layers have a 3x3 filter and stride of 1, and its max pooling layers have a 2x2 filter and stride of 2. The three dense layers after the stack of convolutional/max pooling layers have 4096, 4096, and 1000 channels, respectively. The final 1000 channel layer is the softmax layer for classification of the images in the ImageNet dataset (G, Rohini 2021). VGG16 was the starting point for the computer vision model built for this project to classify satellite images. Programming for this model was performed using Keras imported from Tensorflow 2.11.0 in a Python coding framework.

The data used in all the experiments performed comes from Canada's website for forest fire mapping and consists of color satellite images with 2 different classes, "Wildfire" and "No Wildfire". The dataset contains 53% of images from the Wildfire class and 47% of images from the No Wildfire class. A total of 4,285 images were used, and the data was divided into a set of

3,025 training images, 630 validation images, and 630 test images. Each image in the satellite image dataset is a color image in 2D arrays with 350x350 pixels and three channels for red, green, and blue. Each pixel of color red, green, and blue is assigned a value in the range from 0 to 255.

After loading the dataset (train, validation, and test data), images were augmented using the ImageDataGenerator and were flipped randomly and rotated 75 degrees with a batch size of 256. In addition, images were resized from 350x350 pixels, per the original satellite images, to 224x224 pixels to match the image size of the ImageNet dataset that the VGG16 network was pre-trained on.

The models were built using the Sequential class in Keras and consist of the VGG16 model architecture followed by additional layers to fine-tune the model for the specified classification problem. For most experiments, the layers of VGG16 were frozen, i.e. unable to be trained, with the only trainable layers following after the VGG16 architecture. For one experiment, only the first 15 layers of VGG16 were frozen. For each model in this assignment, the output layer is always a dense layer, preceded by a flatten layer, with 2 nodes, one for each image class in the dataset, and Softmax activation function, in most cases. For compiling the model, adam, categorical cross entropy, and accuracy were used for optimizer, loss, and metrics. The learning rate was also varied between 0.00001 and 0.01. Five different experiments were performed with the VGG16 pre-trained network to optimize architecture and hyperparameters for the best output. The experiments performed are outlined in Table 1.

Table 1. VGG16 Pre-Trained Neural Network Experiments

Experiment	Model Description
1	VGG16 w/ no trainable layers + Flatten layer + Dense Layer w/ 2 nodes and default activation; LR = 0.00001
2	VGG16 w/ no trainable layers + Dense layer w/128 nodes and relu activation + Dense layer w/128 nodes and relu activation + Flatten layer + Dense Layer w/ 2 nodes and softmax activation; LR = 0.00001

Experiment	Model Description
3	VGG16 w/ no trainable layers + Dense layer w/128 nodes and relu activation + Dense layer w/128 nodes and relu activation + Flatten layer + Dense Layer w/ 2 nodes and softmax activation; LR = 0.01
4	VGG16 w/ no trainable layers + Dense layer w/128 nodes and relu activation + Dense layer w/128 nodes and relu activation + Dense layer w/128 nodes and relu activation + Flatten layer + Dense Layer w/ 2 nodes and softmax activation; LR = 0.01
5	VGG16 w/ top 15 layers not trainable + Dense layer w/128 nodes and relu activation + Dense layer w/128 nodes and relu activation + Flatten layer + Dense Layer w/ 2 nodes and softmax activation; LR = 0.01

Models were trained on the 3,025 images, with 630 images used for the validation data. Each training utilized early stopping with a patience value of 3, monitoring the validation accuracy. Epochs were set to 200 for each experiment. The five models shown in Table 1 were evaluated to determine which model performed best. Metrics used to evaluate the models include, train, test, and validation accuracy and loss, model accuracy and loss visualizations, and confusion matrix.

Results

A first evaluation was performed on each of the experiments by comparing the train and validation accuracy and loss and test accuracy after training the model through a certain number of epochs. The accuracy and loss data for each experiment is presented in Table 2.

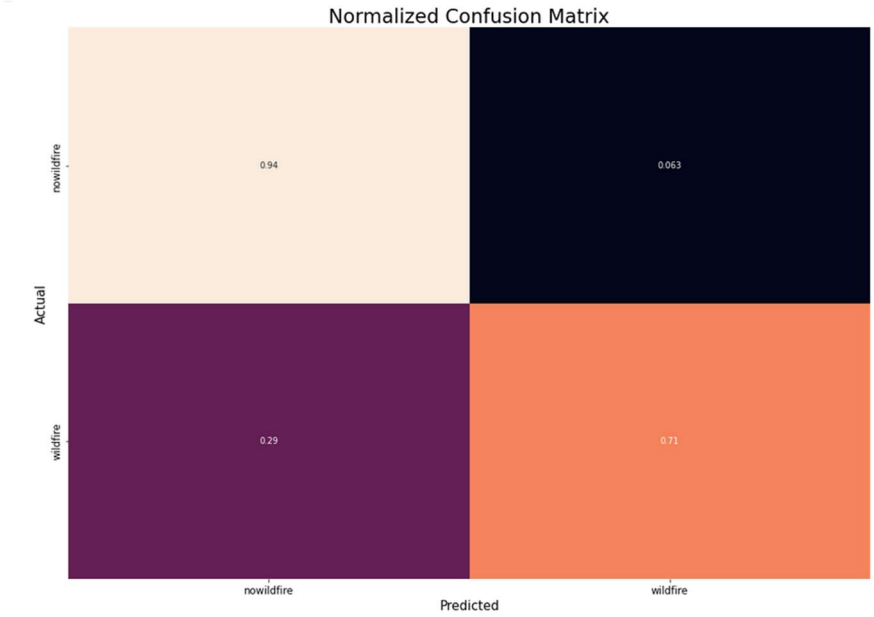
Table 2. Train and Validation Accuracy and Loss and Test Accuracy

Experiment ID	Train Accuracy	Train Loss	Validation Accuracy	Validation Loss	Test Accuracy
1	83.2%	0.50	84.6%	0.41	81.3%
2	92.2%	0.20	91.9%	0.21	92.7%
3	94.4%	0.16	94.4%	0.16	93.3%
4	92.2%	0.21	93.3%	0.18	93.3%
5	90.4%	0.26	91.2%	0.23	92.9%

Experiment 1 was meant to be a baseline model with the simplest fine tuning of the trained VGG16 architecture; all the VGG16 layers were made to be untrainable, so the weights from training on the ImageNet dataset were kept. After the pre-trained VGG16 architecture, a flatten layer was added followed by a dense output layer with two nodes, one for each class of image. Overall, the model performs well with a test accuracy of 81.3%. The train accuracy and

validation accuracy have a difference of only 1.4%, which indicates that the model fits the data well. Based on the confusion matrix, the model performed well when classifying images with no risk of wildfire (No Wildfire class), with an accuracy of 94% correctly classified. However, the model had a more difficult time classifying images with wildfire risk (Wildfire class), with 71% correctly classified. See Figure 2 for the Experiment 1 confusion matrix.

Figure 2. Confusion Matrix Experiment 1

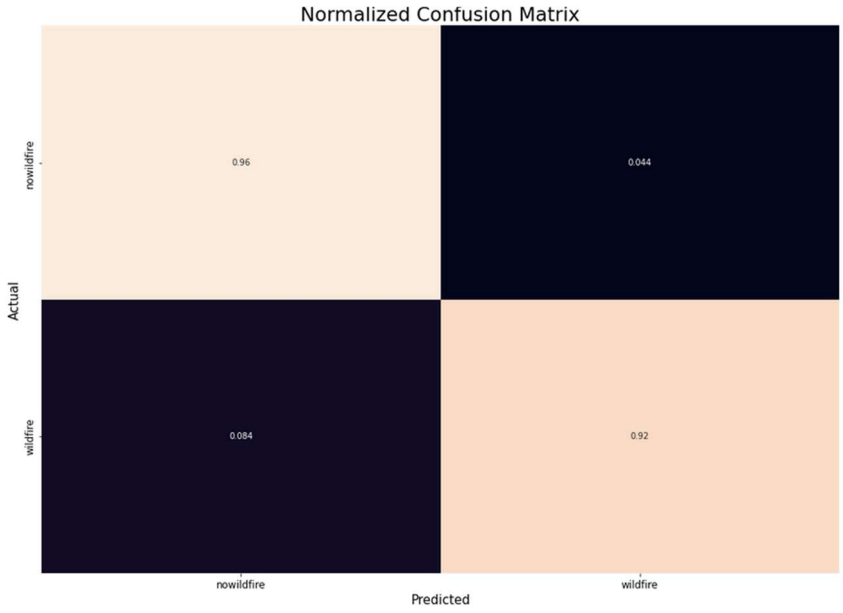


The goal of Experiment 2 was to improve the test accuracy of the model for the satellite image dataset for detecting wildfire risk and to improve the accuracy of Wildfire image classification. For this experiment, two dense layers with 128 nodes and ReLU activation were added after the pre-trained VGG16 architecture and before the flatten and dense 2 node output layer. The output layer activation was specified as Softmax. These additional dense layers add depth to the network and also more gradually step down the number of nodes from 1000 nodes in the last dense layer of VGG16 to the 2 nodes of the output layer of the wildfire model. This model performed much better than the model in Experiment 1, with a test accuracy of 92.7%,

and had perfectly balanced classification between Wildfire and No Wildfire classes, with 93% accuracy for each. The model fit the data well with 0.3% difference between train and validation accuracy. However, since the learning rate was set low at 0.00001, the model took a very long time to train, even with just the few trainable layers.

The goal of Experiment 3 was to decrease training time of the model while keeping the accuracy high, close to that of Experiment 2. For this experiment, everything about the model was kept the same as that of Experiment 2, but the learning rate was increased to 0.01 in order to decrease the training time of the model. The model for Experiment 3 trained about 7 times faster than the model for Experiment 2 with comparable performance. The overall test accuracy was 93.3%, and based on the confusion matrix, the model was able to classify Wildfire images with 96% accuracy and No Wildfire images with 92% accuracy. A slight imbalance in the classification is observed, again in favor of the No Wildfire class, but both are above 90%, so the performance is still very good. Figure 3 shows the confusion matrix for Experiment 3. The test and validation accuracies were the same, indicating a well-fitting model.

Figure 3. Confusion Matrix Experiment 3

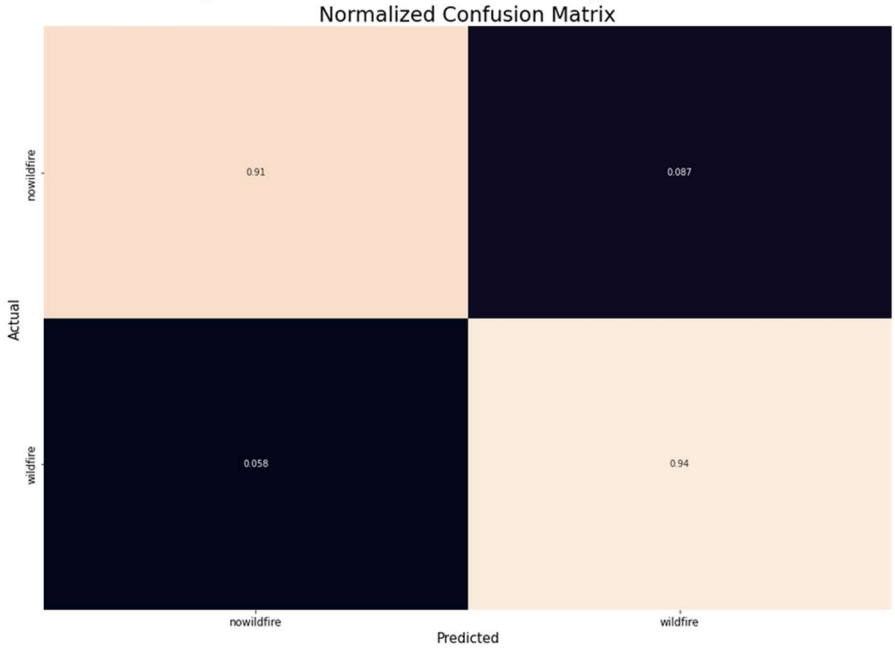


The goal of Experiment 4 was to attempt to improve the test accuracy above 93%. Before performing Experiment 4, it was apparent that a higher test accuracy would be difficult to achieve because the accuracy of the model is already very high. The VGG16 model did not achieve above 93% accuracy on the ImageNet dataset classification. An third dense layer with 128 nodes and ReLU activation was added before the flatten layer, and everything else was kept the same compared to Experiment 3. This additional layer resulted in a slightly faster trained model than Experiment 3 and roughly the same test accuracy of 93.3%. There was still a small imbalance in classification with 96% accuracy of the No Wildfire class and 91% accuracy of the Wildfire class. The train and validation accuracies have a 1.1% difference indicating a good fit of the model to the data. Train and validation losses were slightly higher than those for Experiment 3.

For Experiment 5, some layers of VGG16 were made trainable to determine the impact on the model accuracy. For Experiments 1 – 4, all pre-trained layers of VGG16 were frozen and therefore not trainable with the wildfire dataset. Weights from the ImageNet dataset were kept and used. For Experiment 5, only the first 15 layers of VGG16 were made untrainable, so the last block of CNN layers was trainable (Per Figure 1: Conv 5-1, Conv 5-2, Conv 5-3, and Pooling). Since the top layers of the model extract more general features, these can be frozen while the bottom layers of the model are made trainable to extract features that are more specific to the relevant dataset. In addition, one of the dense 128 node layers with ReLU activation was removed from the model since it did not improve model accuracy, therefore there were two of these dense layers after the VGG16 architecture and before the flatten layer. The learning rate stayed at 0.01. The model for Experiment 5 performed comparably to those of Experiments 2 – 4 with a test accuracy of 92.9%. Based on the confusion matrix, the classification imbalance

changed from favoring the No Wildfire class to favoring the Wildfire class, with 91% accuracy and 94% accuracy, respectively. Allowing some of the bottom layers of the VGG16 architecture to be trained on the wildfire dataset took additional time compared to freezing all the layers in the VGG16 model. Figure 4 shows the confusion matrix for Experiment 5.

Figure 4. Confusion Matrix Experiment 5



Visualizations of model loss and accuracy and confusion matrix for all of the experiments can be found in the Appendices.

Conclusions

The goal of this assignment was to develop a neural network model for image classification for the identification of geographic areas at risk of wildfire. Data used to train and test the model came from Canada's website for forest fire mapping and were extracted from satellite images and put into a format suitable for deep learning and modeling on Kaggle (Aaba 2023). The data available contains satellite images in the classes of "Wildfire" with 22,710 images and "No Wildfire" with 20,140 images. Each image is color and 350x350 pixels. Data

were split into training, validation, and test datasets of 30,025 training images, 6,300 validation images, and 6,300 test images, however only 10% of available data (3,025 training images, 630 validation images, and 630 test images) was used for this project because of limitations on computing power. Experiments were conducted using Python programming framework with pre-trained model VGG16 in Keras with variations made to model architecture and hyperparameters.

The models were built using the Sequential class in Keras and consist of the VGG16 model architecture followed by additional layers to fine-tune the model for the specified classification problem. For most experiments, the layers of VGG16 were frozen, i.e. unable to be trained, with the only trainable layers following after the VGG16 architecture. For one experiment, only the first 15 layers of VGG16 were frozen. For each model in this assignment, the output layer is always a dense layer, preceded by a flatten layer, with 2 nodes, one for each image class in the dataset, and Softmax activation function, in most cases. For compiling the model, adam, categorical cross entropy, and accuracy were used for optimizer, loss, and metrics. The learning rate was also varied between 0.00001 and 0.01. Five different experiments were performed with the VGG16 pre-trained network to determine which architecture had the best output.

Experiment 1 was meant to be a baseline model with the simplest fine tuning of the trained VGG16 architecture; all the VGG16 layers were made to be untrainable, so the weights from training on the ImageNet dataset were kept. After the pre-trained VGG16 architecture, a flatten layer was added followed by a dense output layer with two nodes, one for each class. Overall, the model performs well with a test accuracy of 81.3%. For Experiments 2 – 5, the model was fine-tuned to improve the performance and test accuracy. Additional dense layers were added, the learning rate was increased, and some layers of the VGG16 model were made

trainable on the wildfire dataset. Adding dense layers did improve the test accuracy of the model to approximately 93%. Increasing the learning rate helped the model to train much faster but did cause the classification to become slightly imbalanced between Wildfire and No Wildfire classes. Allowing the last several layers of the VGG16 architecture to be trainable did not change the test accuracy but it did shift the classification imbalance from favoring the No Wildfire class to favoring the Wildfire class.

Overall, the model with the best performance was the model from Experiment 3. This model had the highest accuracies and lowest losses in conjunction with best fit to the data with a test accuracy of 93.3%. However, slight imbalance in classification did occur in favor of the No Wildfire class, but both classes were able to be identified with over 90% accuracy. In addition, the model training time was decreased sevenfold by increasing the learning rate. It is recommended to move forward with the model from Experiment 3 for classification of new satellite images and identification of geographic areas at risk of wildfire.

References

1. Aaba, Abdelghani. “Wildfire Prediction Dataset (Satellite Images).” Kaggle, February 1, 2023. https://www.kaggle.com/datasets/abdelghaniaaba/wildfire-prediction-dataset.
2. Boesch, Gaudenz. “A Complete Guide to Image Classification in 2023.” viso.ai, January 1, 2023. https://viso.ai/computer-vision/image-classification/.
3. G, Rohini. “Everything You Need to Know about VGG16.” Medium. Medium, September 23, 2021. https://medium.com/@mygreatlearning/everything-you-need-to-know-about-vgg16-7315defb5918.
4. “How Artificial Intelligence Revolutionized Computer Vision: A Brief History.” Motion Metrics, May 16, 2019. https://www.motionmetrics.com/how-artificial-intelligence-revolutionized-computer-vision-a-brief-history/.
5. Mihajlovic, Ilija. “Everything You Ever Wanted to Know about Computer Vision. Here's a Look Why It's So Awesome.” Medium. Towards Data Science, April 25, 2019. https://towardsdatascience.com/everything-you-ever-wanted-to-know-about-computer-vision-heres-a-look-why-it-s-so-awesome-e8a58dfb641e.
6. “VGG-16: CNN Model.” GeeksforGeeks. GeeksforGeeks, January 10, 2023. https://www.geeksforgeeks.org/vgg-16-cnn-model/.

Appendices

Figure A1. Loss and Accuracy Experiment 1

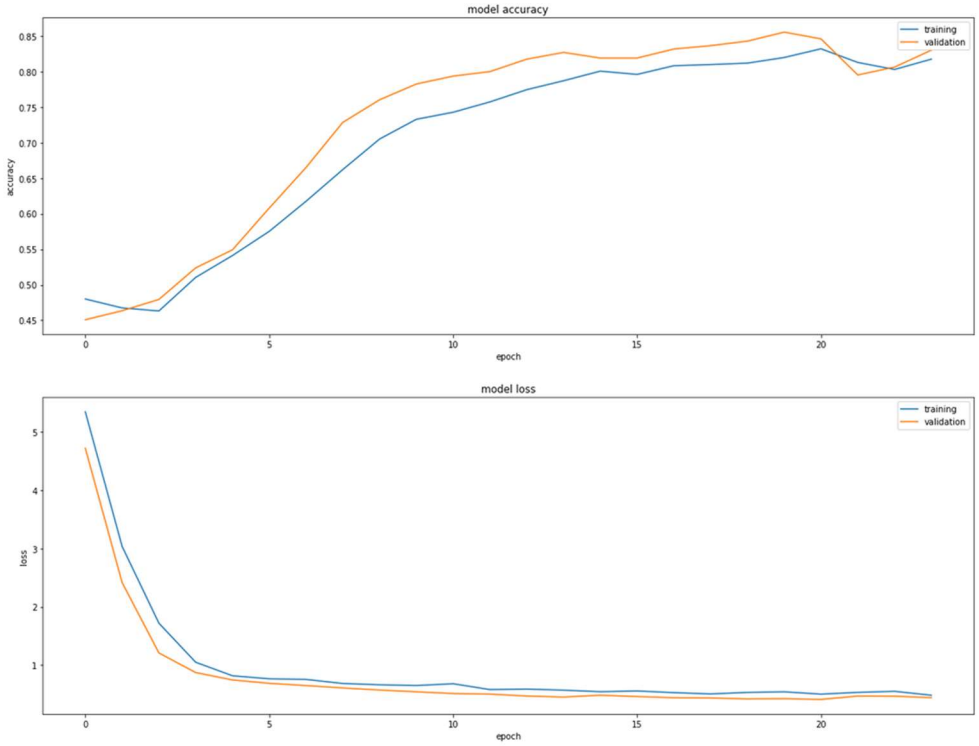


Figure A2. Confusion Matrix Experiment 1

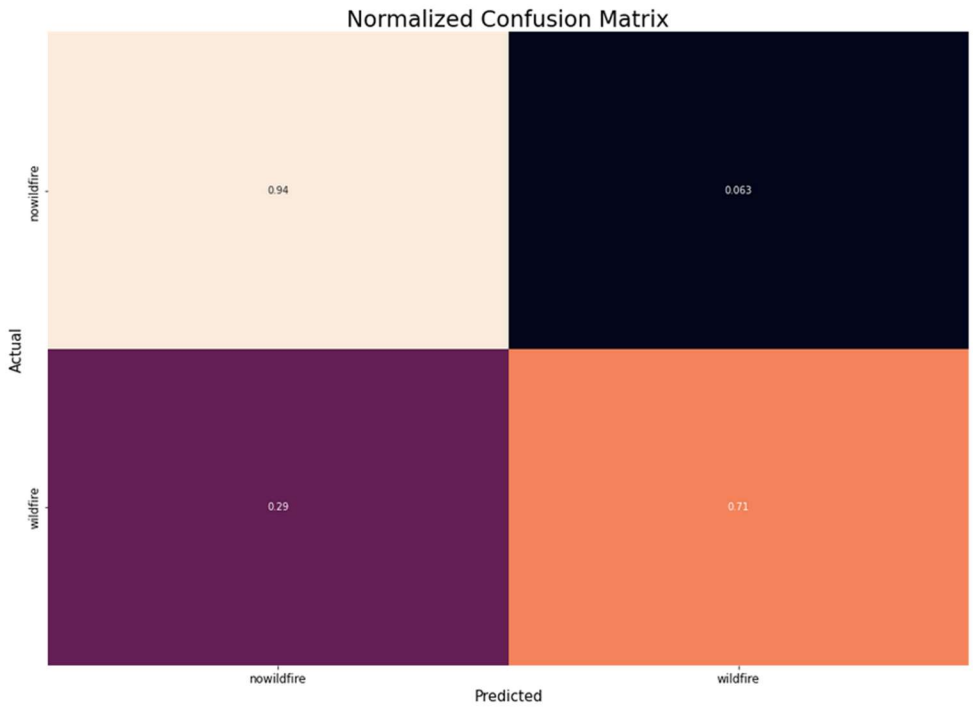


Figure A3. Loss and Accuracy Experiment 2

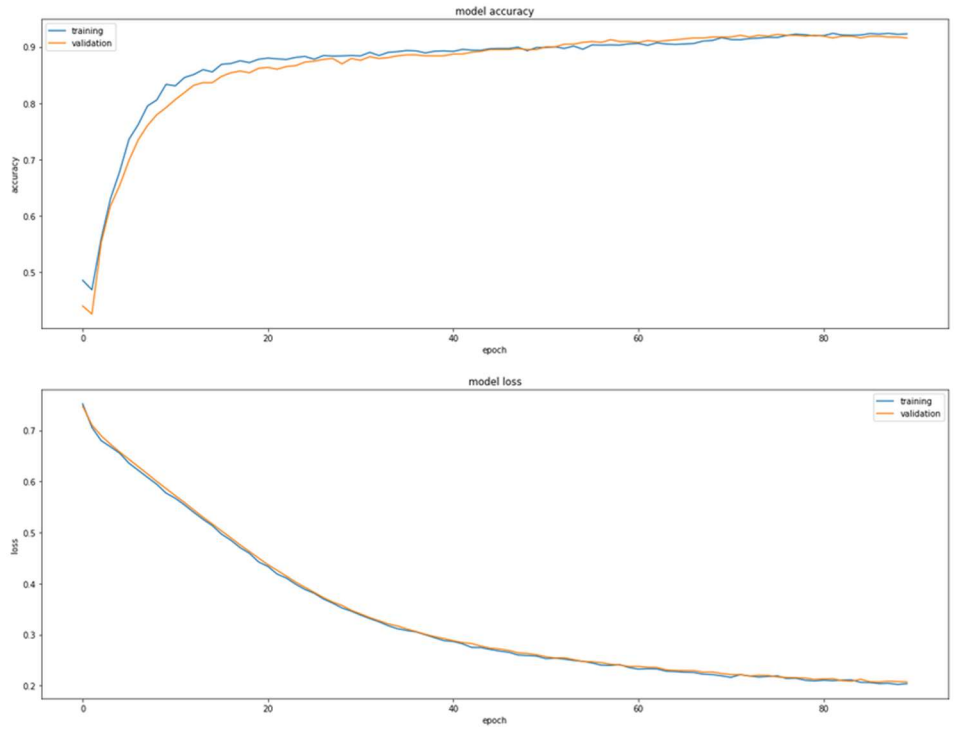


Figure A4. Confusion Matrix Experiment 2

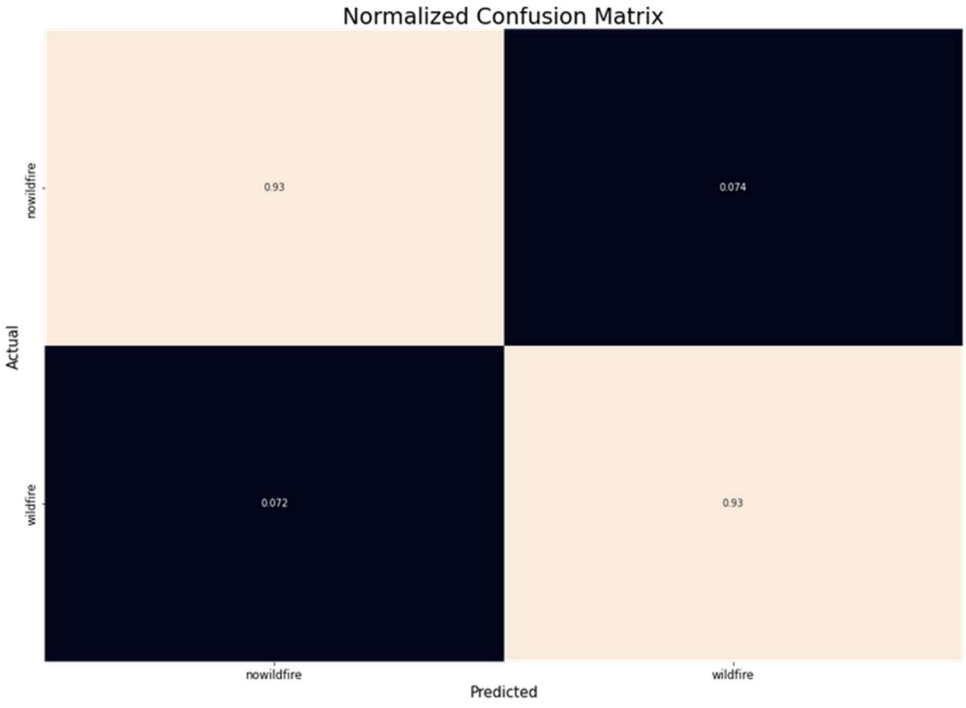


Figure A5. Loss and Accuracy Experiment 3

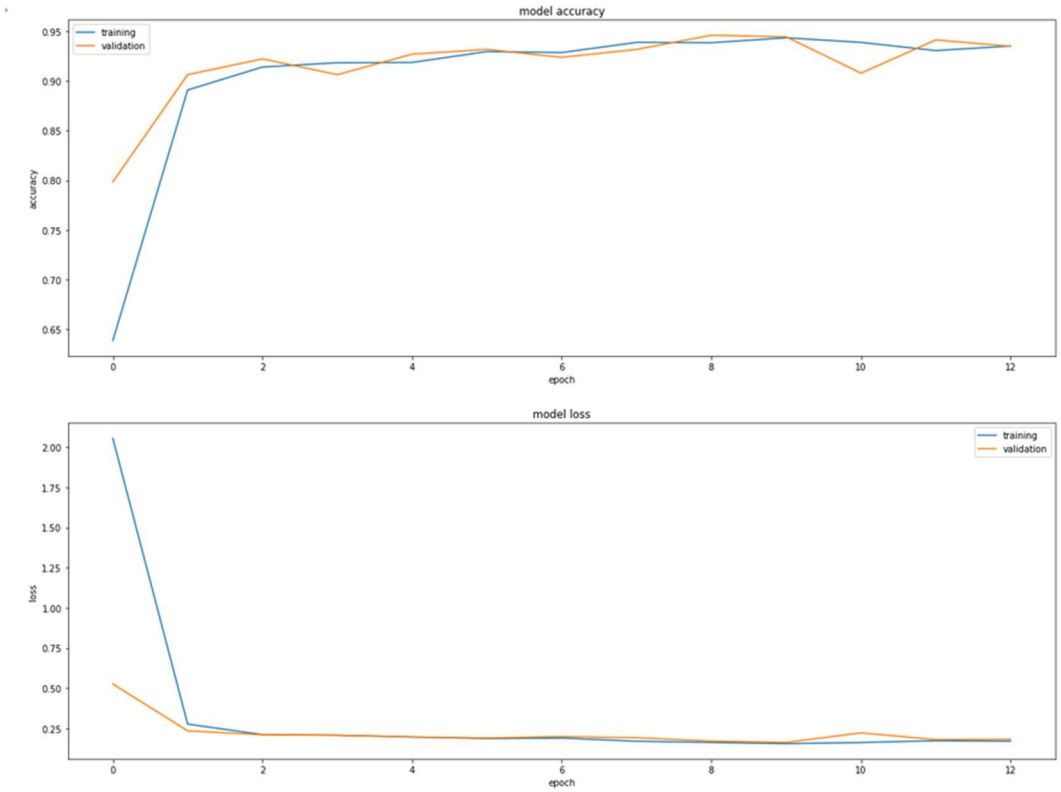


Figure A6. Confusion Matrix Experiment 3

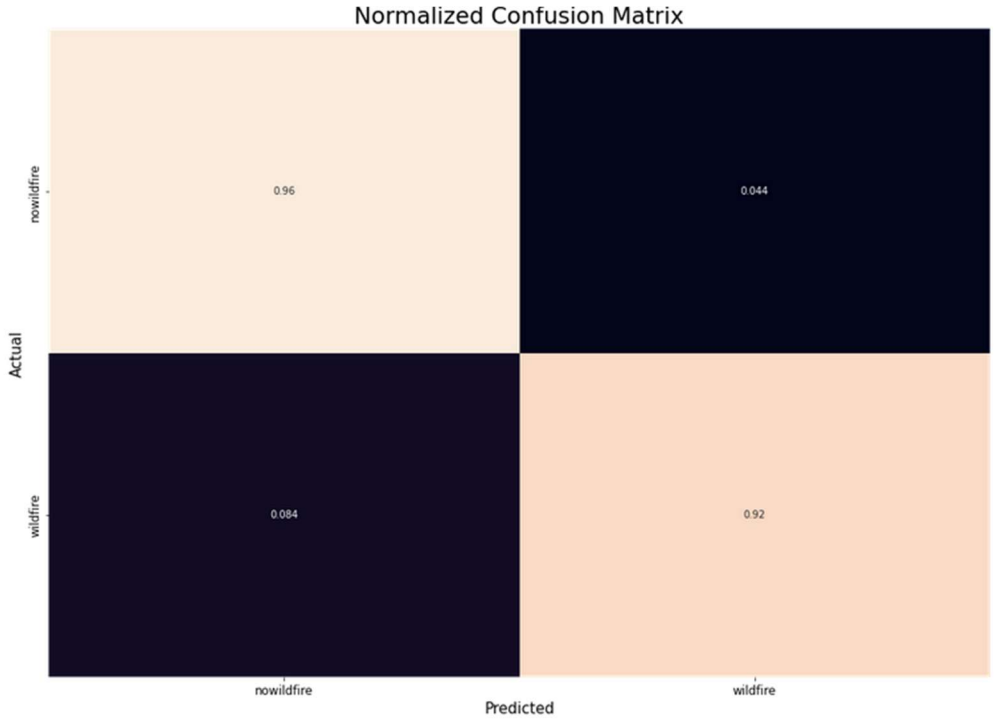


Figure A7. Loss and Accuracy Experiment 4

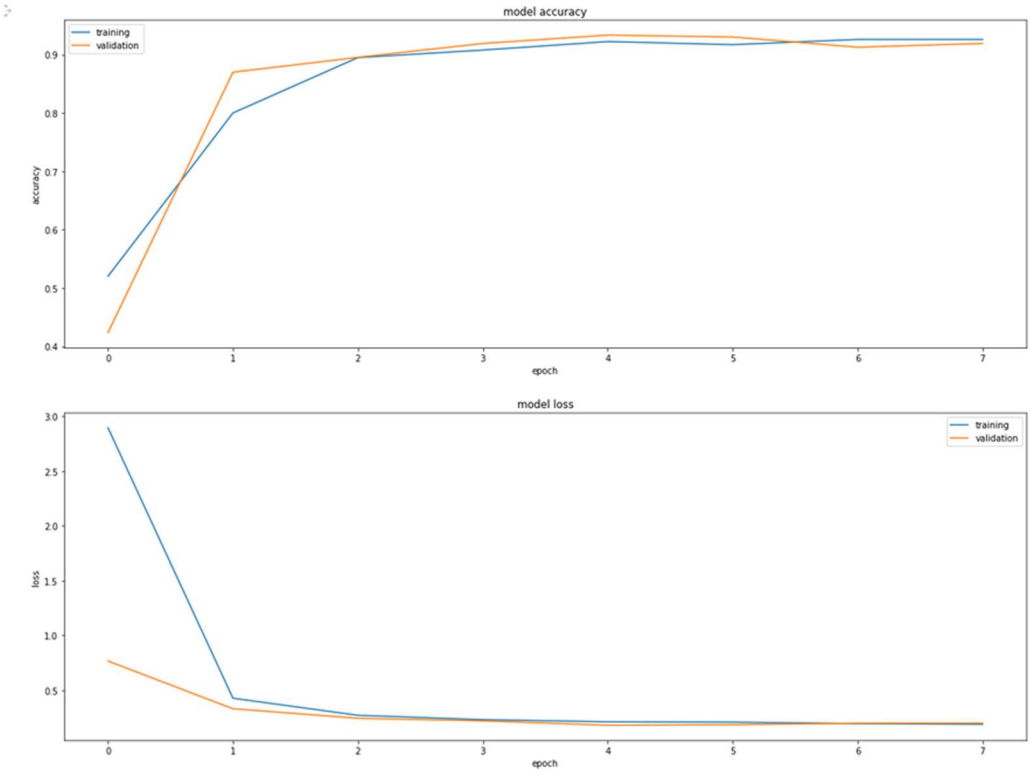


Figure A8. Confusion Matrix Experiment 4

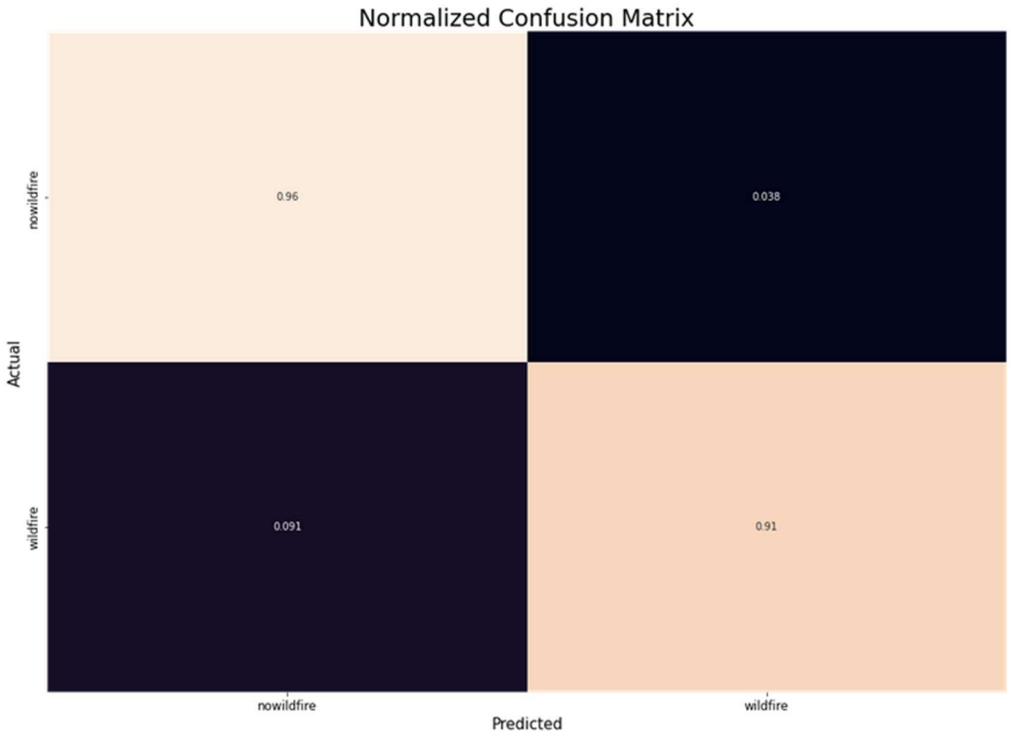


Figure A9. Loss and Accuracy Experiment 5

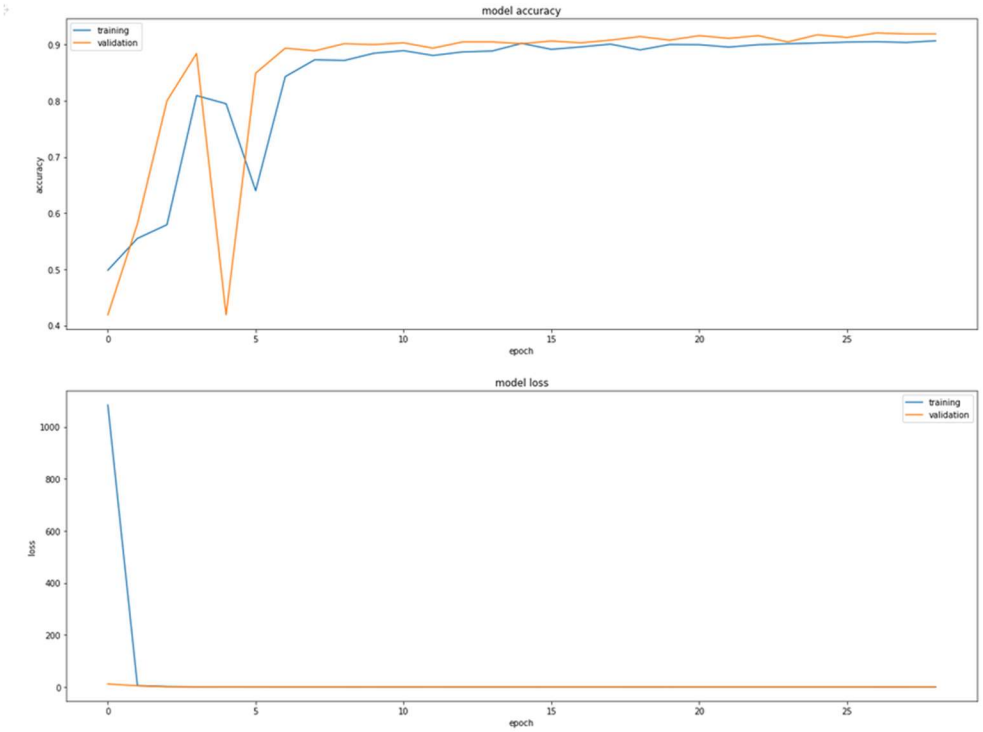


Figure A10. Confusion Matrix Experiment 5