Analysis of popular CNNs on image classification datasets

Paul Norton
Faculty of Science
Western University
London, Ontario
pnorton4@uwo.ca

Nirmal Vettiankal
Faculty of Science
Western University
London, Ontario
nvettian@uwo.ca

Michael Ens
Faculty of Science
Western University
London, Ontario
Mens3@uwo.ca

Bradley Assaly-Nesrallah
Faculty of Science
Western University
London, Ontario
bassalyn@uwo.ca

***Abstract*—Image classification and computer vision is one of the most important topics in the Artificial Intelligence revolution. Being able to identify and classify images and videos is a simple task for a human brain but exceptionally difficult for traditional machine learning algorithms and neural networks. Advances in deep learning such as convolutional neural networks are breaking the problem wide open. Four well-known CNNs are AlexNet, ResNet, VGG and GoogLeNet. We propose that the structure of a deep learning network will affect the performance on different classification tasks. We have tested the performance on three image datasets, SVHN, CIFAR-10 and CelebA. The results of our project have shown that different networks perform best on each dataset. This shows that the architecture of a CNN effects its performance on differing image classification problems.**

***Index Terms*—Image Classification, Neural Network, CNN, AlexNet, ResNet, VGG, GoogLeNet, SVHN, CIFAR-10, CelebA**

I. Introduction

With the rise of modern deep learning overtaking the performance of traditional machine learning algorithms on various problems we have new tools to tackle previously difficult tasks. Image Classification is a vital problem for the development of artificial intelligence due to the large number of parameters and immense amount of data required. Deep learning has been shown to outperform traditional machine learning when the datasets are exceptionally large. This has led to the application of neural networks to solve more complex problems such as image classification. Convolutional Neural Networks, or CNNs, employ the mathematical operation of convolution in at least one of the layers to aid in the performance of image recognition.

Several prominent CNNs such as AlexNet, ResNet, VGG and GoogLeNet have been widely touted for their performance on image classification problems. All of them have distinct architecture in terms of the number of layers, the neurons in each layer and the activation functions used. We want to understand how this architecture effects the performance of image classification in the context of different settings. We test this using several data sets which mimic different image classification problems. We will evaluate the relative performance of each network on the datasets. This will provide us insight into the relationship between CNN architecture and image classification performance. We will first examine related work, followed by a discussion of our data and we will perform an exploratory data analysis. Then we will contextualize four CNNs we are studying by comparing and contrasting their architecture. Finally we will discuss our experimental methods and the results followed by drawing our final conclusions.

II. Related Work

Convolutional Neural Networks have been used in many research papers for various problems of image classification. They typically aim to determine what is the most accurate CNN for any given problem. While some were mostly focused on evaluating the performance of each CNN, others aim to understand which architecture performs the best in different image classification problems [3]. While they were mostly focusing on comparing the performance of CNNs on a given dataset [4]. Our project aims to determine how each CNN performs relative to each other on differing datasets in hopes to understand the relationship between architecture and performance.

However there are a multitude of different image classification problems, a long term theoretical goal is to find an optimal solution to all image classification problems. This could potentially be a CNN with a given architecture or it could be another algorithm entirely. Research papers have aimed to find the optimal CNN for the ImageNet dataset [1]. Our paper seeks to compare the performance of different architectures over multiple datasets.

III. Data

The problem to be solved by this project is how various CNN architectures are able to classify images in different data sets. We will be using three distinct data sets for our project - SVHN, CIFAR10 and CelebA. Each of these three datasets represents a different type of image recognition problem. The SVHN dataset is about digit classification from afar, the CIFAR10 dataset is about classification of ten various objects or animals, meanwhile CelebA is facial attribute regognition.

The SVHN dataset [6] (shown “Fig. 1”), is a real world image dataset similar to MNIST with 100000 digits. Each digit is a 32x32 pixel RGB real world image from a street view house number. The dataset is subset into 73257 digits for training and 26032 digits for testing. There are 10 classes, 1 for each digit. The classes for this datset are mutually exclusive.

Fig. 1. Sample of SVHN images [6]

The CIFAR10 dataset [2] (shown "Fig. 2") is a real world image dataset of 60000 images from 10 classes corresponding to real world entities such as planes, cats, and ships. The dataset is subset into 50000 training images and 10000 test images with an equal proportion of each class. Each image is a 32x32 pixel RBG image. The classes for this dataset are completely mutually exclusive.

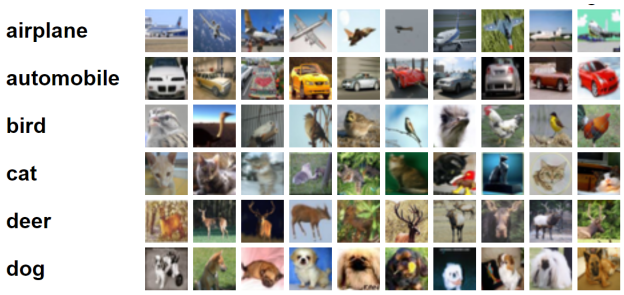

Fig. 2. Sample of CIFAR10 classes [2]

The CelebA dataset [5] (shown "Fig. 3") consists of 200,000 celebrity images of size 178x218, in RGB format. Each image has multiple targets: identity, bounding box and 40 binary attributes. For this study, we focused on the binary classification only.

IV. Exploratory Data Analysis

Exploratory data analysis is the critical step of performing initial investigations on the original dataset so as to discover patterns and prepare for the correct direction of the further data cleaning process. We use several statistical and visual approaches to understand the data. Each dataset has a distinct structure, so we will discuss the differences in depth as follows.

A. Target

In our image classification or binary attribute classification tasks the goal is to classify each image into its correct target (label). For the SHVC dataset the target is the 10 classes representing the 10 possible digits, for CIFAR-10 dataset the target is 10 lables corresping to image classes, meanwhile for the CelebA dataset there are multiple targets: identity,

Fig. 3. Sample of CelebA attributes [5]

bounding box and 40 binary attributes. For this study, we focused on the binary classification only.

B. Image Type

In our image classification or binary attribute classification tasks, the type of image in each dataset will impact the effectiveness of each CNN. For the SVHN dataset, the image is a 32x32 RGB image, for CIFAR-10 dataset the image is a 32x32 RGB image, meanwhile for the CelebA dataset the images are 178x218 RGB images. The difference in size of the image could impact the effectiveness of each CNN.

C. Target Distribution

In our classification tasks when we train our CNNs the distribution of the targets could affect the effectiveness of our training. We consider the distribution and symmetry of each data sets targets as follows. SVHN and CIFAR-10 both have a uniform distribution of each label, equivalently stated that there are equal training samples with each label, meanwhile the distribution of CelebA is not uniform as the quantity of each target varies throughout the training set.

V. Convolutional Neural Networks

Convolutional Neural Networks are a class of deep neural networks typically applied to analyzing images. They are regularized versions of multilayer perceptron neural networks. Convolutional networks are fully connected and consist of typical layers as well as convolutional layers which filter the data. The architecture of a CNN can vary greatly in terms of its layers, layer size, input and output size, convolutional layers, and activation functions. There are many well known CNNs which we will use throughout our paper.

A. AlexNet

One of the most well known convolutional neural networks is AlexNet [1], which was designed by Alex Krizhevesky and won the ImageNet 2012 competition. The architecture of AlexNet consists of eight layers, the first five of which

are convolutional, followed by max-pooling layers, and finally the last three are fully connected layers. The layers consist of 4096 neurons for both fully connected layers followed by 1000 output neurons, which we have resized to fit our output spaces. The activation functions used are the non-saturating ReLU activation function and the softmax for the output layer. The structure of AlexNet is shown in "Fig. 4".

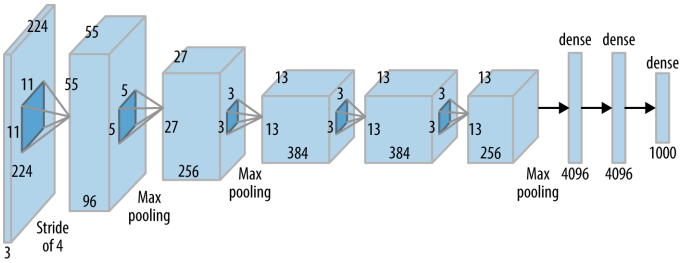

Fig. 4. Architecture of AlexNet [1]

B. ResNet

Another famous convolutional neural network is ResNet [7], designed by Kaiming He and won the ImageNet 2015 competition. The core idea of ResNet is introducing a residual element to the layer where a layer skips one or multiple layers which aimed to tackle the vanishing gradient problem. ResNet was a very deep neural net, and achieved greater accuracy on classification problems then the earlier AlexNet. The activation functions used are the non-saturating ReLU activation function and the softmax for the output layer. The structure of ResNet is shown in "Fig. 5".

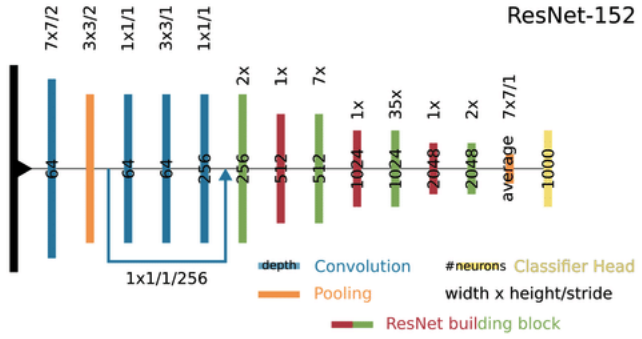

Fig. 5. Architecture of ResNet [7]

C. VGG

The next well known convolutional neural network we will consider is VGG [8] designed by Simonyan and Zisserman. VGG is 16 layers deep and seeks to make improvements over AlexNet by replacing large kernel sized filters with multiple 3x3 kernel sized filters. VGG has convolutional layers followed by max pooling, with three fully connected layers with 4096 neurons each. This architecture uses the ReLU activation function for the convolution and fully connected layers and the softmax for the output layer. The structure of VGG is shown in "Fig. 6".

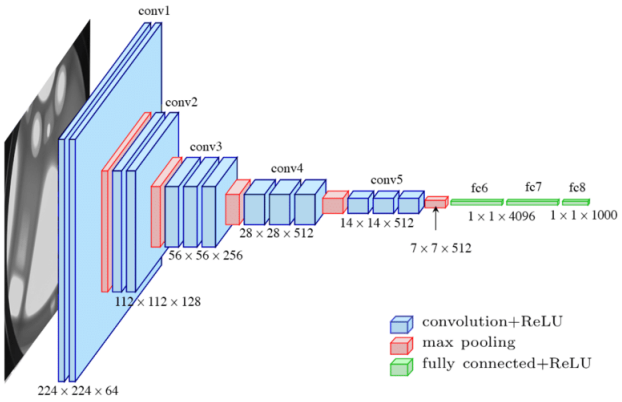

Fig. 6. Architecture of VGG [8]

D. GoogLeNet

The final convolutional neural network we will consider is GoogLeNet [9] developed by researchers at Google. This CNN is considered an inception network and was a breakthrough for the field of deep learning. The architecture of the network is 22 layers deep with 27 pooling layers included. There are 9 inception modules that are stacked linearly. At the end of each inception module each is connected to the global average pooling layer. The structure of GoogLeNet is shown in "Fig. 7".

type	patch size/ stride	output size	depth	#1×1	#3×3 reduce	#3×3	#5×5 reduce	#5×5	pool proj	params	ops
convolution	7×7/2	112×112×64	1							2.7K	34M
max pool	3×3/2	56×56×64	0								
convolution	3×3/1	56×56×192	2		64	192				112K	360M
max pool	3×3/2	28×28×192	0								
inception (3a)		28×28×256	2	64	96	128	16	32	32	159K	128M
inception (3b)		28×28×480	2	128	128	192	32	96	64	380K	304M
max pool	3×3/2	14×14×480	0								
inception (4a)		14×14×512	2	192	96	208	16	48	64	364K	73M
inception (4b)		14×14×512	2	160	112	224	24	64	64	437K	88M
inception (4c)		14×14×512	2	128	128	256	24	64	64	463K	100M
inception (4d)		14×14×528	2	112	144	288	32	64	64	580K	119M
inception (4e)		14×14×832	2	256	160	320	32	128	128	840K	170M
max pool	3×3/2	7×7×832	0								
inception (5a)		7×7×832	2	256	160	320	32	128	128	1072K	54M
inception (5b)		7×7×1024	2	384	192	384	48	128	128	1388K	71M
avg pool	7×7/1	1×1×1024	0								
dropout (40%)		1×1×1024	0								
linear		1×1×1000	1							1000K	1M
softmax		1×1×1000	0								

Fig. 7. Structure of GoogLeNet [9]

VI. Methods

The project aims to build and compare four image classification models performance over three different data sets. The first thing that we do is take each CNN and train it using each dataset's training subset. Then we evaluate the performance of each CNN on the dataset using the loss and accuracy metrics. We can use this performance to compare the relative performance of each CNN for the different image classification tasks.

A. Training with Backpropagation

Each of the CNNs needs to be trained in order to effectively perform classification of images. The first thing we do is to

iterate over the training set for each dataset. On each iteration we do a forward pass which obtains our output from the model, then we use the backpropagation algorithm to modify the weights to minimize our loss function. We use the cross entropy loss function to train our networks. Cross Entropy Loss: $J = -\frac{1}{N}\sum_{i=1}^{N} y_i \cdot log(p(y_i)) + (1-y_i) \cdot log(1-p(y_i))$

We do this for for multiple epochs until the training loss is minimized. This is the point we consider the CNN to be trained, so we are done.

B. Testing the Models Performance

Once we have trained our model we can evaluate the performance by iterating over the test set and comparing the predictions of the model with the labels of the test set. We obtain our accuracy measure as follows:
Accuracy = correct predictions/ total predictions
Now we have obtained a metric with which to compare the performance of each CNN on our datasets.

VII. Experimental Results

A. CelebA Binary Attributes

The CelebA dataset contains a mixture of subjective (e.g. "attractive") and objective ("wearing sunglasses") binary classification attributes. The models were asked to identify all 40 in a single pass. Model accuracy is shown in "Fig. 8".

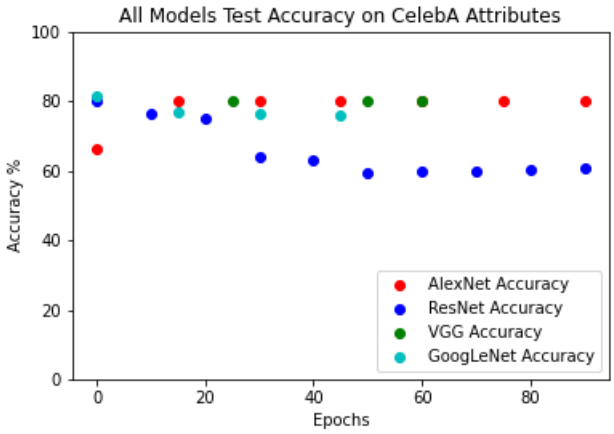

Fig. 8. Model Accuracy on CelebA Attributes

1) AlexNet: After 90 epochs of training, AlexNet finished tied for the highest accuracy alongside VGG-16, at 79.9% accuracy on the testing dataset. The final model size was 223 MB.

2) GoogLeNet: GoogLeNet was left to train for 45 epochs. It strikes a good balance between accuracy and size, with a final accuracy of 76.1% and a model size of 47 MB.

3) ResNet-18: ResNet was also trained for 90 epochs, which is far more than standard, and as a result we begin to see overfitting around 20-30 epochs. After 20 epochs, the model had a 74.8% accuracy. The intermediary models were preserved in order to "rewind" after overfitting. The final model size was 44 MB.

4) VGG-16: VGG-16 is a very large model that was slow to train. On a 1080, it took approximately 12 hours to train, but ended up tied for the second highest accuracy at 79.9%. Its model size was 525 MB.

B. CIFAR-10 Classification

The CIFAR-10 dataset contains a mixture of images of various objects with labels such as "cat" or "airplane". The models were asked to identify what class was in the image from 10 possible classes in a single pass. Model accuracy is shown in "Fig. 9".

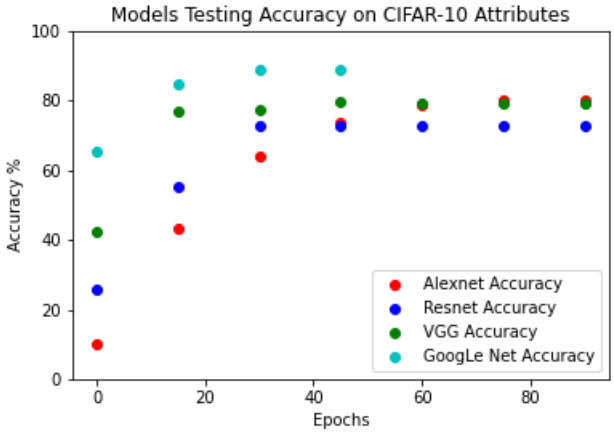

Fig. 9. Model Accuracy on CIFAR-10 classification

1) AlexNet: After 90 epochs of training, AlexNet finished tied for the second highest accuracy, at 79.9% accuracy on the testing dataset. AlexNet did not look like it was overfitting as it was gaining accuracy up till the cutoff at 90 epochs. The final model size was 223 MB.

2) GoogLeNet: GoogLeNet was left to train for 45 epochs as it seemed to overfit at around the 30 epoch. It's small size and highest score with a final accuracy of 88.9%, as well as low training needed led it to be the best model for this data set. A final model size of 47 MB was recorded.

3) ResNet-18: ResNet was also trained for 90 epochs, which is far more than standard, and as a result we begin to see over fitting around 30 epochs. At 30 epochs, the model had a 72.8% accuracy and after went down to 72.7% accuracy. The intermediary models were preserved in order to "rewind" after over fitting. The final model size was 44 MB.

4) VGG-16: VGG-16 is a very large model that was slow to train. On a 2080, it took approximately 9 hours to train, and ended up with the second lowest accuracy at 79.0%. Its model size was 525 MB. This model is probably the worst to use as the size and training time are not worth it compared to other models.

C. SVHN Classification

The SVHN dataset contains real world images of house number from a street view. The models were asked to identify the digit in the image in a single pass. Model accuracy is shown in "Fig. 10".

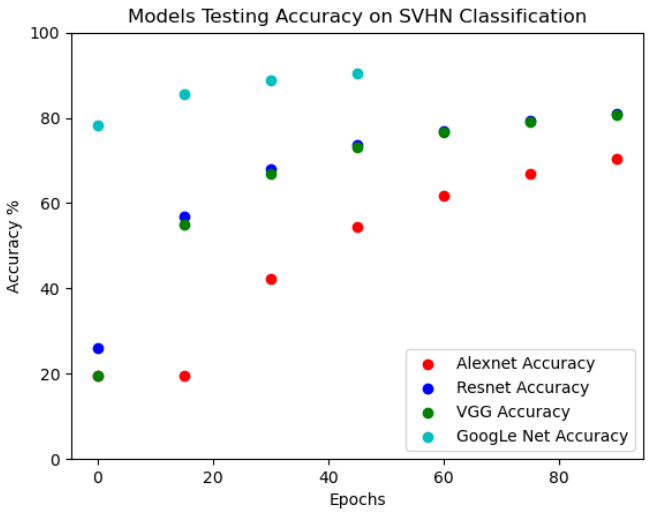

Fig. 10. Model Accuracy on SVHN Classification

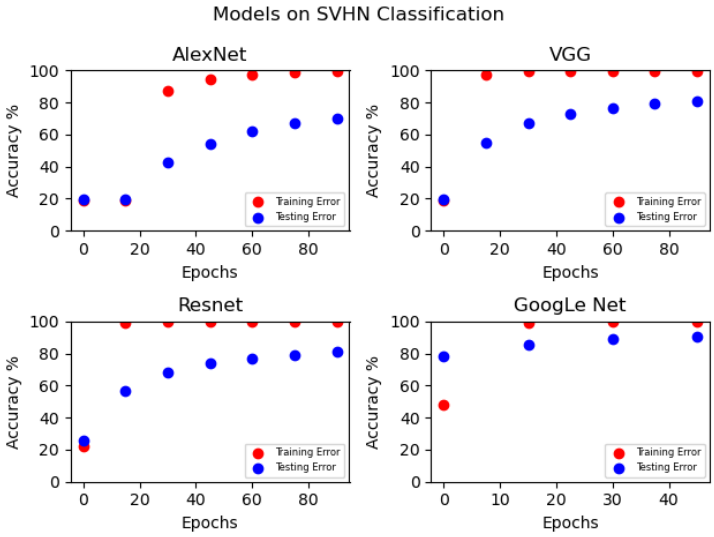

Fig. 11. Training and Testing Accuracy on SVHN Classification

1) AlexNet: After 90 epochs of training, AlexNet finished with the lowest accuracy at 70.3% accuracy on the testing dataset. The final model size was 223 MB.

2) GoogLeNet: GoogLeNet was left to train for 45 epochs. It was both the most accurate and took the least space, with a final accuracy of 90.4% and a model size of 47 MB.

3) ResNet-18: ResNet was also trained for 90 epochs and had a similar result to VGG, with less size used. The testing accuracy was 80.9% and the final model size was 44 MB.

4) VGG-16: VGG-16 is a very large model that was slow to train. On a 2070, it took approximately 9 hours to train and ended with an accuracy of 80.8%. Its model size was 525 MB.

VIII. Conclusions

There was a consistent over-fitting issue with all models trained on the SVHN dataset. The training and testing accuracy discrepancies can be seen in "Fig. 11". The over-fitting is most likely due to the complexity of the model being greater than the task of classifying 10 classes from a 32x32 image. All the models used were intended for 224x224 images belonging to 1000 classes which is a far more complex problem.

Similarly on the CelebA dataset, our capable models (AlexNet, GoogLeNet, VGG) all seemed to hit their maximum around 80%. Given the subjective nature of some of the labels in the dataset (e.g. "attractive", "oval face"), we feel this is an acceptable accuracy. On more objective facial feature datasets, this accuracy could improve and we could see a separation in performance between models.

On the CIFAR10 dataset, the AlexNet model was still improving at epoch 90. The other models seemed to overfit at around epoch 30 and stayed consistent at their accuracy. Most models seemed to stay around low to high 70s, but GoogLeNet was the best at 89%. This proved that GoogLeNet was the best for the CIFAR10 dataset.

Comparing the performance of our four CNNs on the datasets we can see that the GoogLeNet had the highest accuracy for the SVHN and Cifar-10, meanwhile AlexNet and VGG had the highest performance for the CelebA dataset. Thus our of our four CNNs there was not one dominant architecture for all of our datasets. Each one had varying performance for different datasets. The complex architecture of GoogLeNet allowed it to perform strongly on the two image classification datasets, however it did not perform as well on the binary attribute classification.

References

[1] A. Krizhevsky, I. Sutskever, and G. E Hinton, "ImageNet Classification with Deep Convolutional Neural Networks," May 10, 2013. Accessed: April 11, 2021. [Online]. Available: https://papers.nips.cc/paper/2012/file/c399862d3b9d6b76c8436e924a68c45b-Paper.pdf

[2] A. Krizhevsky, "Learning Multiple Layers of Features from Tiny Images," April 8, 2009. Accessed: April 11, 2021. [Online]. Available: https://www.cs.toronto.edu/~kriz/learning-features-2009-TR.pdf

[3] S.H. Shabbeer Basha, Shiv Ram Dubey, Viswanath Pulabaigari, Snehasis Mukherjee, "Impact of fully connected layers on performance of convolutional neural networks for image classification," April 8, 2009. Accessed: April 11, 2021. [Online]. Available: https://www.sciencedirect.com/science/article/abs/pii/S0925231219313803

[4] C. Luo, X. Li, L. Wang, J. He, D. Li and J. Zhou, "How Does the Data set Affect CNN-based Image Classification Performance?," July 15, 2018. Accessed: April 11, 2021. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/8599448

[5] Liu, Ziwei and Luo, Ping and Wang, Xiaogang and Tang, Xiaoou, "Proceedings of International Conference on Computer Vision (ICCV)," December, 2015. Accessed: April 11, 2021. [Online]. Available: https://arxiv.org/abs/1411.7766

[6] Y. Netzer, T. Wang, A. Coates, A. Bissacco, B. Wu, A. Y. Ng, "Reading Digits in Natural Images with Unsupervised Feature Learning", 2011. Accessed: April 11, 2021. [Online]. Available: http://ufldl.stanford.edu/housenumbers/nips2011_housenumbers.pdf

[7] K. He, X. Zhang, S. Ren, J. Sun, "Deep Residual Learning for Image Recognition", December 10, 2015. Accessed: April 11, 2021. [Online]. Available: https://arxiv.org/abs/1512.03385

[8] K. Simonyan, A. Zisserman, "Very Deep Convolutional Networks for Large-Scale Image Recognition," September 4, 2014. Accessed: April 11, 2021. [Online]. Available: https://arxiv.org/abs/1409.1556

[9] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, A. Rabinovich, ”Going Deeper with Convolutions“, September 17, 2014. Accessed: April 11, 2021. [Online]. Available: https://arxiv.org/abs/1409.4842