Using Deep Learning to Analyze the Driveability of Autonomous Vehicles on Unmarked Terrain

Arjun Gupte[1], Chandreyi (Zini) Chakraborty[2], and Charles Zhu[3]

Basis Independent Silicon Valley, agupte@ucsb.edu[1], Oak Ridge High School, cchakraborty@ucsb.edu[2], Monta Vista High School, czhu742@ucsb.edu[3]

I. ABSTRACT

Autonomous vehicles have tremendous applications in off-road exploration situations, so it is crucial to analyze the driveability (relative safety) of the terrains that they will be traversing. This paper sheds light on the relatively unexplored domain of determining the driveability of a given terrain in hopes of ensuring the safety of the occupants and vehicle during travel. Traditional research focuses on terrain classification on a small scale, identifying textures, objects (i.e branches and rocks), and colors, but is limited in its applications, especially with regards to autonomous driving. Our method consists of a deep convolutional neural network capable of simultaneously classifying the terrain in an image as well as its relative driveability. This neural network is built upon a modified and pre-trained ResNet18 architecture. We created a customized dataset of hundreds of images belonging to seven different terrains, which we manually labeled and fed into the model. Finally, we compared the performances of different pre-trained model architectures and analyzed their results for each terrain and driveability category.

II. INTRODUCTION

Modern-day autonomous vehicles have been specialized to navigate through cluttered and primarily paved roads, but are yet to be optimized for traversing rugged and off-road terrain. Developing a deterministic model to classify unmarked terrain and assess its safety would create numerous opportunities for advancements in large-scale autonomous vehicle development and testing. For example, an autonomous vehicle that is able to analyze its ability to drive over the terrain in its path would be able to make more intelligent decisions about whether it should reroute to avoid dangerous obstacles.

Prior research [1], [2], [3] focuses solely on the identification of terrain, and it is often limited to small samples or zoomed-in images that are not suitable for the real-life scenarios autonomous vehicles encounter. Additionally, minimal work that assesses the driveability of a vehicle on a given terrain has been carried out. This is quite detrimental to the development of autonomous vehicles as determining the safety of the surface a vehicle is driving on is crucial for ensuring the safety of the passengers.

Our research seeks to establish a more robust method of analyzing the safety of the terrain by surveying the landscape as a whole (as opposed to discrete object detection [4], [5], [6]). To do this, we curated a custom dataset consisting of a variety of different driveable and not-driveable terrains, giving us a wide variety of images to train our neural networks on. We then developed three convolutional neural networks that are capable of classifying terrain, assessing its driveability, and doing both tasks simultaneously on two parallel threads. As a result of this structure, we are able to analyze the outputs of the networks and recommend the highest performing technique.

III. METHODS / METHODOLOGY

A. Neural Network Models

The base neural network architecture that we decided to use for our research was a ResNet18 model pre-trained on ImageNet. We chose this architecture due to its relatively small number of hidden layers and modest depth. Using a model that is fairly shallow is crucial for our small dataset to avoid over-analyzing and over-fitting. As we sought to perform three tasks with this architecture, we created three variations of it: one for solely classifying terrain, another for solely assessing the driveability, and the third for combining both of these tasks into one neural network. For the two independent models, we modified the last Linear Layer of ResNet18 so that the output would correspond to the number of categories we created for each task (seven for terrain and two for driveability). For the third variation, we modified the last Linear Layer to accommodate for one of the tasks and created another Linear Layer in parallel to accommodate for the other task. The general architecture of the independent terrain model can be seen in **Figure 1.1**, the independent driveable model in **Figure 1.2**, and the combined model in **Figure 1.3**.

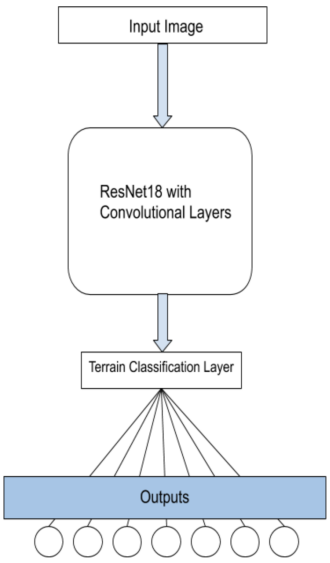


Figure 1.1: Terrain classification model

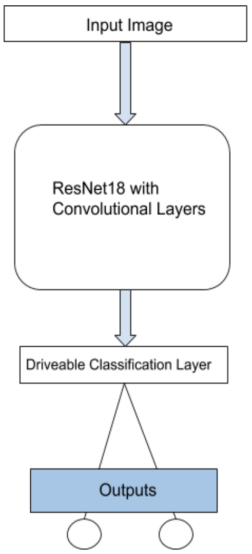


Figure 1.2: Driveable classification model

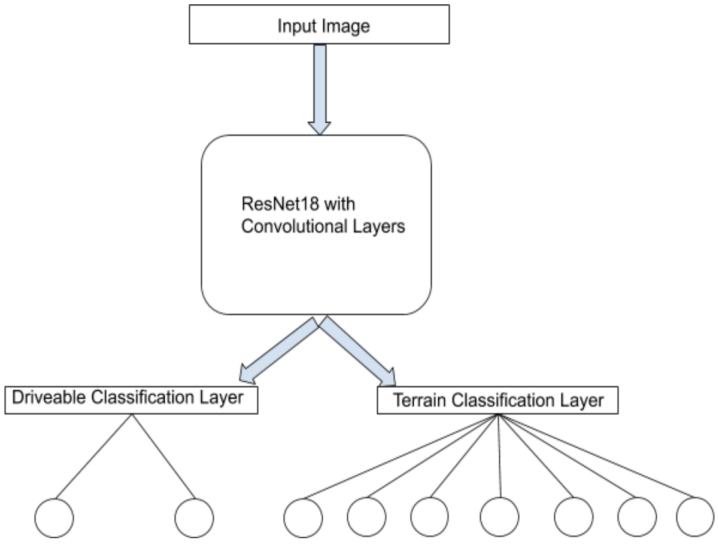


Figure 1.3: Mixed classification model

For all three models, we used the pre-implemented ReLU activation function from the ResNet18 architecture for its simplicity, relatively high performance, and ability to circumvent the "vanishing gradient" problem (unlike other activation functions such as Sigmoid). Furthermore, we continued using the pre-implemented Cross-Entropy Loss Function as seen in (1) from the pre-trained ResNet18 network for all three models.

$$L_{\text{cross-entropy}}(\widehat{y}, \text{y}) = -\sum_{i} y_i log(\hat{\text{y}}_\text{i}) \quad (1)$$

B. Dataset

Our data consists of images we collected from Google Images that holistically represent each terrain we wanted to train the model on. We compiled fifty to one hundred images for each of the seven terrains: dirt, muddy, normal asphalt, rocky, sandy, snowy, and wet. Then, we generated two separate datasets using these images: one that categorizes the data by terrain, and another that categorizes it by driveability. We defined driveability as a binary value - either driveable or not driveable - and determined an image was not driveable if any of the following conditions are met:

1. Dirt: not driveable if a clear path is indistinguishable from surroundings and/or trees/bushes are in the way
2. Muddy: not drivable if thick, wet, or deep segments of mud are present
3. Normal Asphalt: not driveable if deep potholes and/or large cracks are present
4. Rocky: not driveable if sharp and large protruding rocks are present
5. Sandy: not driveable if large sand dunes are present
6. Snowy: (a) not driveable if road and/or tire tracks are not visible, and (b) not driveable if the road is fully covered with snow and lane markings are hidden
7. Wet: not driveable if the road is flooded or water level has risen above tire height.

A few examples of these classifications can be seen in **Figure 2.1**, **Figure 2.2**, and **Figure 2.3**.

Figure 2.1: Sandy, Not Driveable [7]

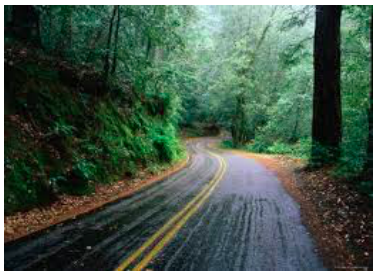


Figure 2.2: Wet, Driveable [8]

Figure 2.3: Rocky, Not Driveable [9]

C. Techniques

To ensure our model is trained sufficiently on all categories, we balanced our dataset to contain an equal number of images from each terrain and driveable category within our train and test folders. Before running the images through our models, we implemented a few data augmentations to increase the randomness in our training

images, hence improving the versatility of our model. Each image received a random-sized crop, a random horizontal flip, a random rotation, and a random change in brightness, contrast, and saturation. A few examples of similar data augmentations can be seen in **Figure 3**. In all three of our models, we used a batch size of 16, ran the model for 20 epochs, and used an optimizer with a learning rate of 0.001 and a momentum of 0.9, as we found these to be the most optimal values to ensure relatively high testing accuracies. Additionally, we implemented a scheduler which decays the learning rate by a factor of 10 every 10 epochs. We then tested various pre-trained model architectures and compared their performance on the dataset.

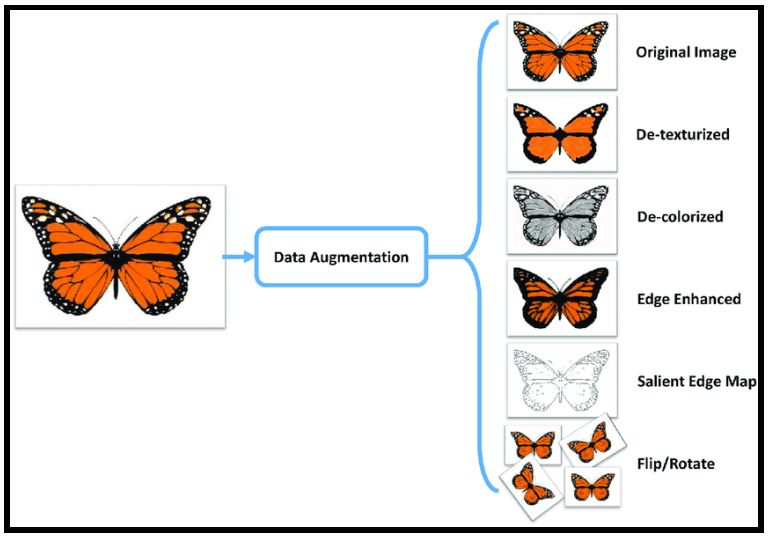


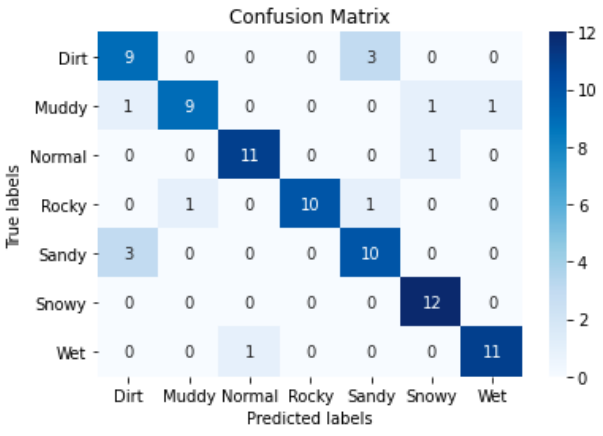
Figure 3: Examples of data augmentations performed on a training image [10]

Our end goal was to run a combined model that is able to train for both terrain and driveability tasks simultaneously. We achieved this by utilizing our previous image folders to create text files (one for training and one for testing) containing a list of images and their respective terrain and driveability labels.

IV. RESULTS / ANALYSIS

A. Terrain Results

This section illustrates the performance of our three models (terrain-only, driveable-only, and mixed) through confusion matrices. Ideally, the diagonal (from the upper left corner to the lower right corner) of the confusion matrix should have the largest values.



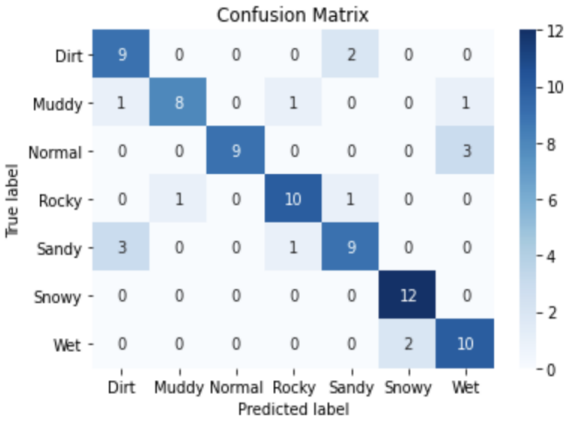


Figure 4.1: Terrain-only model using ResNet18

Figure 4.2: Mixed model using ResNet18

In **Figure 4.1** and **Figure 4.2** we can see that both models do really well at classifying each terrain, nearly getting perfect accuracies. We observed that our model struggled with identifying a few select terrains. The major error in the matrices mostly comes from sandy terrain being predicted as dirt, likely due to the similar colors and textures of these terrains. Another issue was that the model often confused normal, wet, and snowy roads with each other. This is also understandable because the images frequently had asphalt roads with slight variations in brightness, reflectivity, and snowfall that might have been difficult to detect after the images had various features (brightness, contrast, etc.) modified.

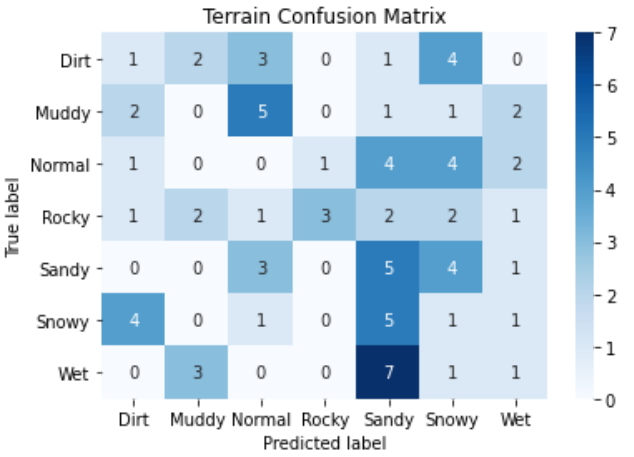


Figure 5.1: Mixed model using base pre-trained ResNet18

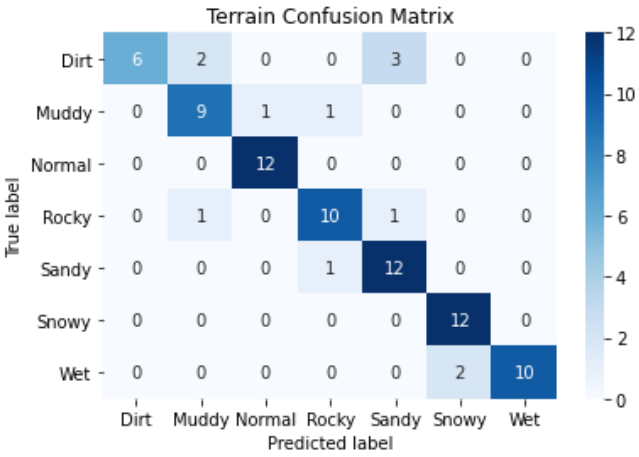
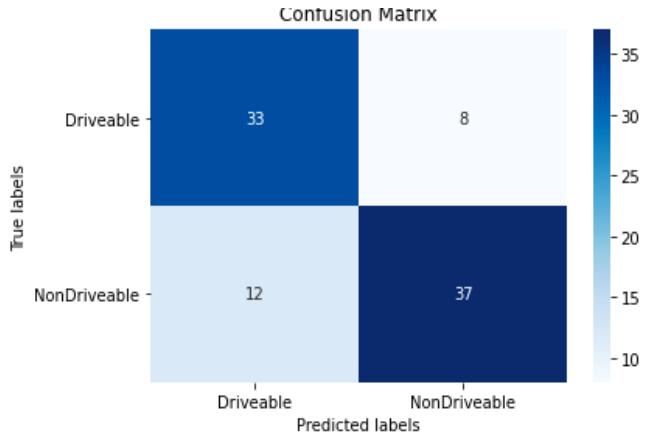


Figure 5.2: Mixed model using VGG16

These matrices indicate that the base ResNet18 model (**Figure 5.1**), trained only on ImageNet, has no form of pattern recognition without prior training on our training dataset. This is because the features that the neural network needs to learn in order to perform well on our dataset naturally vary greatly from those needed for the ImageNet dataset. We then tested several different architectures, the most successful one being VGG16, shown in **Figure 5.2**. For our VGG16 model, we found that dirt roads were often falsely predicted as sandy or muddy. This is likely due to similar color schemes among those three terrains.

B. Driveability Results



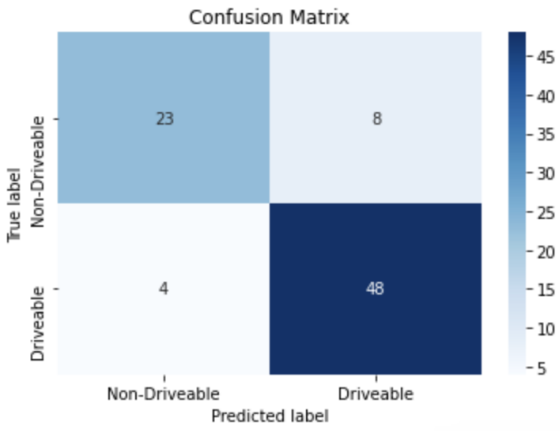


Figure 6.1: Driveable-only model using ResNet18

Figure 6.2: Mixed model using ResNet18

For our driveable-only and mixed models, the results, as shown in the confusion matrices in **Figure 6.1** and **Figure 6.2**, were quite promising. The diagonal going from the upper left of the matrix to the bottom right yielded relatively high values indicating that the models predicted the driveabilities mostly correctly. We hypothesized that our ability to divide the dataset clearly into the two categories along with the models' quickly learned ability to pick up on these patterns were the root causes for these accurate results.

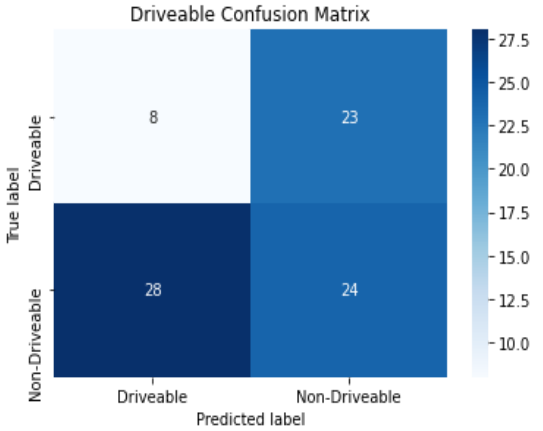


Figure 7.1: Mixed model using base pre-trained ResNet18

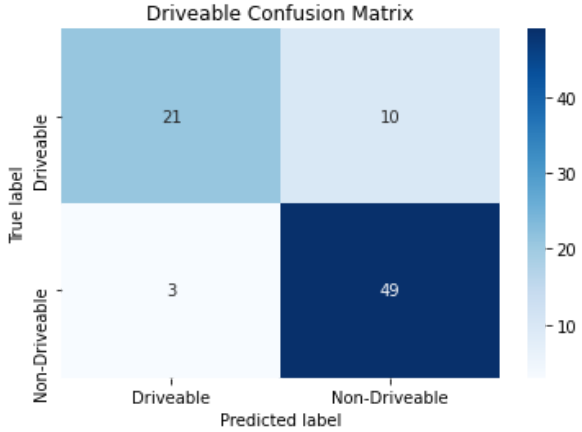


Figure 7.2: Mixed model using VGG16

Similar to the base ResNet18 terrain confusion matrix, the scattered numbers in the base ResNet18 driveable confusion matrix (**Figure 7.1**) show the model's lack of pattern recognition as it has not been trained on our dataset. However, similar to its terrain confusion matrix, the VGG16 model (**Figure 7.2**) shows a much higher accuracy, likely due to it learning some of the definitions that we associated with our images more quickly and confidently.

C. Accuracy Statistics

ResNet18 models	Terrain		Driveability	
	Accuracy	Precision	Accuracy	Precision
Terrain-only model	76.47%	0.764	N/A	N/A
Driveable-only model	N/A	N/A	70.00%	0.725
Mixed model	72.29%	0.7342	83.13%	0.827

Table 1: Terrain and driveability statistics for all three models using ResNet18

As shown in **Table 1,** the results from our terrain classification models using ResNet18 demonstrate that the terrain-only model (76.47%) performed slightly better than the mixed model (72.29%). As the mixed model is attempting to perform two tasks simultaneously, we believe its lower accuracy arose during the combination of

the losses. Specifically, the summing of the two tasks' losses resulted in a net loss that led to the weights being tuned more favorable for one task (in this case driveability).

When analyzing the driveability of a terrain, the mixed model yielded a much higher accuracy than the driveable-only model, surpassing it by 13%. Similar to the justification for the terrain results, there is a chance that the weights were tuned more favorably for the driveability task, leading to the mixed model performing better than the driveable-only model.

Mixed Model Architectures	Terrain Test Accuracies (%)	Driveability Test Accuracies (%)
ResNet18	79.52	80.72
SqueezeNet1.0	43.37	78.31
GoogLeNet	72.29	81.93
DenseNet121	78.31	84.34
VGG16	85.54	84.34
ResNeXt50	81.93	80.72
Wide ResNet50	85.54	84.34

Table 2: Mixed model test accuracies with various architectures

Table 2 shows a comparison of the final test accuracies of the various architectures we tested on the mixed model. As shown in the table, for the task of terrain classification, VGG16 yielded the highest accuracy while DenseNet121, VGG16, and Wide ResNet50 all had the highest final accuracies for the driveability task.

V. DISCUSSION / CONCLUSION

Throughout our research, we focused on determining a robust way to help autonomous vehicles identify different terrains as well as whether or not they are driveable. We accomplished this by developing a complex model that is able to perform both tasks simultaneously with high accuracies. We created a dataset consisting of terrain images gathered from a more holistic perspective (which is much more similar to how autonomous vehicles will receive images in real-life scenarios) as opposed to previous research that utilized zoomed-in images.

Our research could be extended in various directions in the future. One direction could be increasing the total size of our dataset, thereby increasing the versatility of our model, as our current manually-selected dataset contains only around four hundred images. Furthermore, adding more terrain categories (i.e., tundra, swampy, grassy, etc.) to our dataset would be valuable for expanding our model's scope as it would account for a greater variety of real-life conditions. Another extension would be to incorporate images taken from a camera physically mounted to a car to improve the practicality of our model. Similarly, exploring the use of videos instead of static images as a whole is quite appealing as most autonomous vehicles use videos as inputs rather than singular images. Finally, changing our method of classifying the driveability by creating a spectrum of driveability readings as opposed to a binary decision would be an interesting modification to pursue, as our model would then be able to provide more fine-tuned suggestions to the driver.

REFERENCES

[1] T. Selvathai, J. Varadhan and S. Ramesh, "Road and off road terrain classification for autonomous ground vehicle," 2017 International Conference on Information Communication and Embedded Systems (ICICES), Chennai, 2017, pp. 1-3, doi: 10.1109/ICICES.2017.8070724.

[2] F. Ebadi and M. Norouzi, "Road Terrain detection and Classification algorithm based on the Color Feature extraction," 2017 Artificial Intelligence and Robotics (IRANOPEN), Qazvin, 2017, pp. 139-146, doi: 10.1109/RIOS.2017.7956457.

[3] A. Angelova, L. Matthies, D. Helmick and P. Perona, "Fast Terrain Classification Using Variable-Length Representation for Autonomous Navigation," 2007 IEEE Conference on Computer Vision and Pattern Recognition, Minneapolis, MN, 2007, pp. 1-8, doi: 10.1109/CVPR.2007.383024.

[4] N. Vandapel, D. F. Huber, A. Kapuria and M. Hebert, "Natural terrain classification using 3-d ladar data," IEEE International Conference on Robotics and Automation, 2004. Proceedings. ICRA '04. 2004, New Orleans, LA, USA, 2004, pp. 5117-5122, doi: 10.1109/ROBOT.2004.1302529.

[5] Manduchi, R., Castano, A., Talukder, A. et al. Obstacle Detection and Terrain Classification for Autonomous Off-Road Navigation. Autonomous Robots 18, 81–102 (2005). https://doi.org/10.1023/B:AURO.0000047286.62481.1d.

[6] P. Y. Shinzato, D. F. Wolf and C. Stiller, "Road terrain detection: Avoiding common obstacle detection assumptions using sensor fusion," 2014 IEEE Intelligent Vehicles Symposium Proceedings, Dearborn, MI, 2014, pp. 687-692, doi: 10.1109/IVS.2014.6856454.

[7] Lilian. “KANAB: Gems of the Vermillion Cliffs National Monument.” *Lilian Pang*, 16 Nov. 2018, www.lilianpang.com/kanab-vermillion-cliffs-national-monument/.

[8] “'Rain and Wet Roads through the California Forests, California' Photographic Print - Thomas Winz.” *Art.com*, www.art.com/products/p13074765-sa-i2303774/thomas-winz-rain-and-wet-roads-through-the-california-forests-california.htm.

[9] Herkewitz, William. “New Zoom-and-Enhance Technique Brings Mars Into Focus.” *Popular Mechanics*, Popular Mechanics, 15 Feb. 2018, www.popularmechanics.com/space/moon-mars/a20626/zoom-and-enhance-on-mars/.

[10] Kumar, S. (2019, July 21). Data Augmentation Increases Accuracy of your model-But how? Retrieved July 24, 2020, from https://medium.com/secure-and-private-ai-writing-challenge/data-augmentation-increases-accuracy-of-your-model-but-how-aa1913468722.

[11] He K., Zhang X., Ren S., Sun J. (2015, December 10). Deep Residual Learning for Image Recognition. Retrieved July 24, 2020, from the arXiv database. https://arxiv.org/abs/1512.03385.

[12] Landola F., Han S., Moskewicz M., Ashraf K., Dally W., Keutzer K. (2016, February 24). SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and <0.5MB model size. Retrieved July 24, 2020, from the arXiv database. https://arxiv.org/abs/1602.07360.

[13] Szegedy C., Liu W., Jia Y., Sermanet P., Reed S., Anguelov D., Erhan D., Vanhoucke V., Rabinovich A. (2014, September 17). Going Deeper with Convolutions. Retrieved July 24, 2020, from the arXiv database. https://arxiv.org/abs/1409.4842.

[14] Huang G., Liu Z., Maaten L., Weinberger K. (2016, August 25). Densely Connected Convolutional Networks. Retrieved July 24, 2020, from the arXiv database. https://arxiv.org/abs/1409.1556.

[15] Simonyan K., Zisserman A. (2014, September 4).Very Deep Convolutional Networks for Large-Scale Image Recognition. Retrieved July 24, 2020, from the arXiv database. https://arxiv.org/abs/1409.1556.

[16] Xie S., Girshick R., Dollár P., Tu Z., He K. (2016, November 16). Aggregated Residual Transformations for Deep Neural Networks. Retrieved July 24, 2020, from the arXiv database. https://arxiv.org/abs/1611.05431.

[17] Zagoruyko S., Komodakis N. (2016, May 23). Wide Residual Networks. Retrieved July 24, 2020, from the arXiv database. https://arxiv.org/abs/1605.07146.

[18] *ImageNet*, www.image-net.org/index.

ACKNOWLEDGEMENTS

We would like to thank Jedrzej (Jacob) Kozerawski, our professor, and Aiwen Xu, our teaching assistant, for the tremendous help and support they gave us over the course of our research. Additionally, we would like to thank Dr. Lina Kim and the team that organized the UC Santa Barbara Summer Research Academies for making this research opportunity possible.

AUTHOR CONTRIBUTION STATEMENT

All authors (A.G., C.C., and C.Z.) conceived the research idea, gathered the datasets, built the models, and analyzed the results.