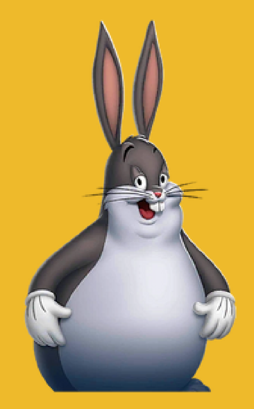


Adapted from [30]

LEPUS CLASSIFIER

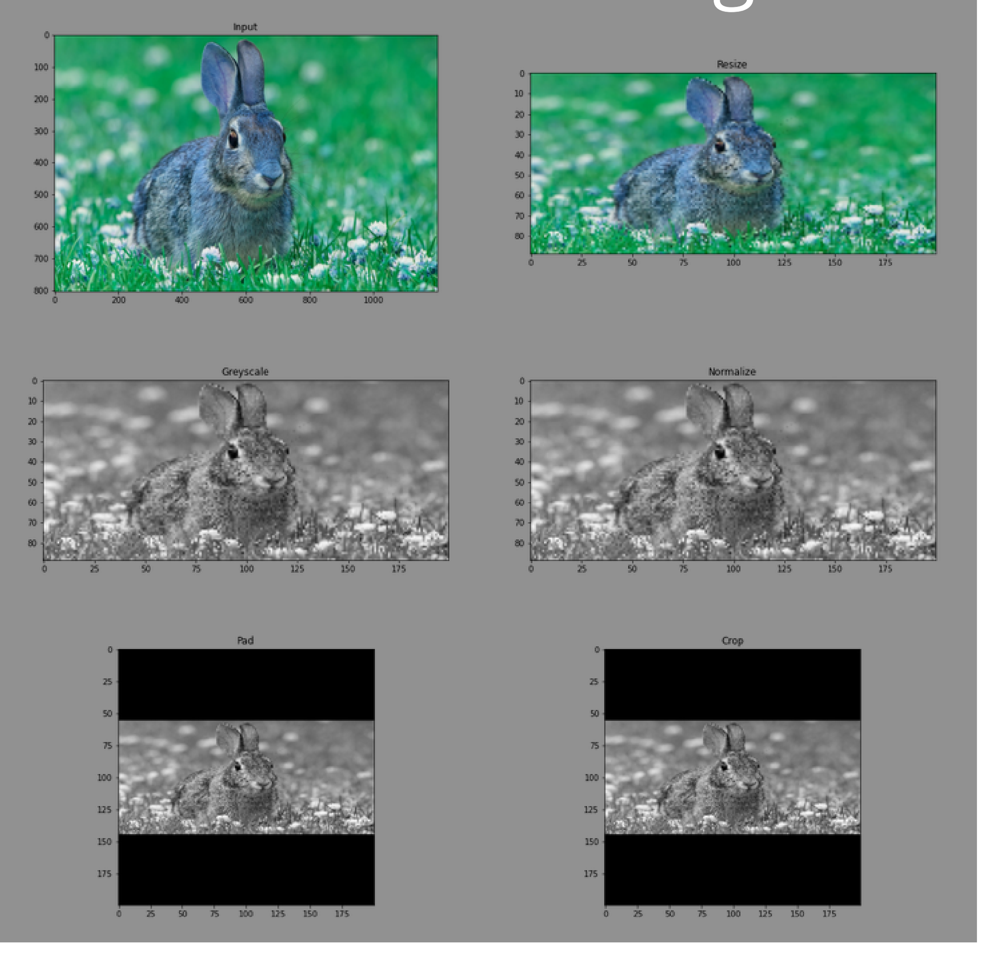
Ben, Urmzd, Keelin

State-of-the-art image classifiers are typically training on hundreds of thousands of images and require extensive computing power. In this report, we examine methods to improve performance on a CNN without the need for large data sets and specialized hardware. Using 85 images of two species from the Lepus genus, we demonstrate that optimal image classifier architectures are still limited by the quantity of data they are trained with, especially when images have highly complex feature sets.

Introduction

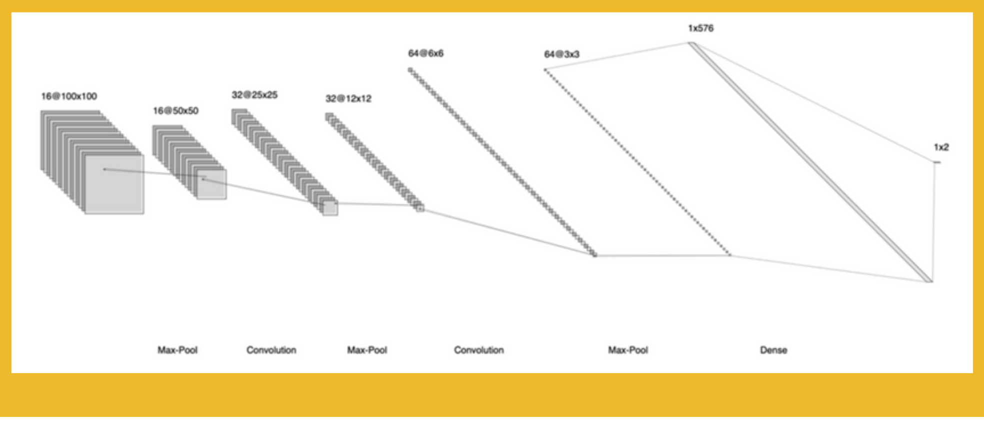
In the scope of this project, we consider the supervised learning problem of image classification using neural networks to differentiate between Eastern cottontail rabbits and European hares. We design and train CNN architectures using a dataset of images collected via web scraping in our approach to this single label, multi-class classification problem. Furthermore, we conduct experiments to explore the design process of training a CNN using foundational image processing techniques, different model layers, activation functions, and hyper-parameter tuning to better understand how neural networks behave when applied to this task.

Data Processing



Model Architecture

Prior to running our experiments, we determined that we needed to develop a baseline model from which iterative experimentation could occur. The aim was to create a model that provide mediocre results, but provided room for improvement through our testing. The resultant model visible below provided a baseline training, validation and test accuracy of 0.91, 0.69 and 0.53 respectively. From this foundation, we were better positioned to evaluate the impact of experimental models.



Experiments & Analysis

Dropout

We observed that the baseline model architecture with a drop rate of 0.5 (Drop02) provided the highest validation accuracy at 0.77, but with a slightly lowered testing accuracy at 0.47.

Pooling

After running the model with average pooling (Pool01), we found that it drastically increased validation accuracy to 0.85, though it slightly reduced testing accuracy to 0.47.

Convolutions

We found that the experimental model with kernel size 3 and stride length of 1 (Convolution02) performed the best, based on an increased validation accuracy of 0.76 and constant test accuracy of 0.53.

K-Fold Cross Validation

The optimal number of folds to use during training was three. Folds greater than three resulted in the unbalanced class being over-shadowed. Folds less than three had the potential for the unbalanced class to be missing entirely.

Optimizers

SGD outperformed all the optimizers. Due to the small initial learning rate, adaptive algorithms such as Adam were unable to make significant enough changes to warrant an improved in performance.

Name	Test Accuracy (Mean)	Train Accuracy	Validation Accuracy
K-Fold/3	0.647058845	1	0.590909064
SGD	0.470588237	0.963636339	0.769230783
Width01	0.470588237	1	0.769230783
Depth02	0.470588237	0.909090936	0.692307711
Convolution02	0.529411793	1	0.769230783
Pool01	0.470588237	0.690909088	0.846153855
Drop02	0.470588237	0.818181813	0.769230783
Tanh_activations_exp	0.647058845	1	0.615384638
batch_2_exp	0.529411793	1	0.769230783

Batch Size

The optimal batch size for training our baseline model comprises of 2 images. The small batch size allows the network to converge faster based on gradient-learning [27].

Activation Functions

The most balanced performance was yielded by Tanh activation, with with training, testing, and validation accuracies of 1.0, 0.65, and 0.62, respectively.

Depth

We found that both the addition of a convolution and max pooling layer and the further addition of another dropout layer (Depth02) did not improve the model's accuracy, but instead led to further overfitting.

Width

The model with a doubled quantity of filters (Width01) performed with a slightly reduced test accuracy of 0.47, while training and validation accuracy increased to 1 and 0.77 respectively.

Conclusion

From our experimental findings, we have yielded promising results that would serve as a solid foundation for further expansion. Going forward, we would need to expand beyond the toy dataset of 85 images in order to build a more robust system. This is dependent on greater access to computational resources and time. Thus far, we have received valuable indications of the impact on model performance by several critical aspects of the model; however, better differentiation between hares and rabbits requires more feature learning from a larger quantity of data.

References

[1] N. Sharma, V. Jain, and A. Mishra, “An analysis of convolutional neural networks for image classification,” Procedia Computer Science, vol. 132, pp. 377–384, 2018, international Conference on Computational Intelligence and Data Science. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1877050918309335
[2] T. Guo, J. Dong, H. Li, and Y. Gao, “Simple convolutional neural network on image classification,” in 2017 IEEE 2nd International Conference on Big Data Analysis (ICBDA), 2017, pp. 721–724.
[3] K. Gurney, “An introduction to neural networks,” 1997.
[4] R. Grosse, “Lecture 3: Multilayer perceptrons,” University of Toronto Computer Science, 2019.
[5] B. Jena, G. K. Nayak, and S. Saxena, “Convolutional neural network and its pretrained models for image classification and object detection: A survey,” Concurrency and Computation: Practice and Experience, vol. 34, no. 6, p. e6767, 2022. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1002/cpe.6767
[6] C. Seger, “An investigation of categorical variable encoding techniques in machine learning: binary versus one-hot and feature hashing,” 2018.
[7] R. Grosse, “Lecture 9: Convolutional networks,” University of Toronto Computer Science, 2019.
[8] A. Amidi and S. Amidi. [Online]. Available: https://stanford.edu/~shervine/teaching/cs-230/ cheatsheet-convolutional-neural-networks
[9] J. Lederer, “Activation functions in artificial neural networks: A systematic overview,” arXiv preprint arXiv:2101.09957, 2021. [10] I. Goodfellow, Y. Bengio, and A. Courville, Deep Learning. MIT Press, 2016, http://www. deeplearningbook.org.
[11] “Softmax regression.” [Online]. Available: http://deeplearning.stanford.edu/tutorial/supervised/ SoftmaxRegression/
[12] A. R. Draelos, “Multi-label vs. multi-class classification: Sigmoid vs. softmax,” Sep 2019. [Online]. Available: https://glassboxmedicine.com/2019/05/26/classification-sigmoid-vs-softmax/
[13] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Kopf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala, “Pytorch: An imperative style, high-performance deep learning library,” in Advances in Neural Information Processing Systems 32, H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, Eds. Curran Associates, Inc., 2019, pp. 8024–8035. [Online]. Available: http://papers.neurips. cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf
[14] “Nllloss.” [Online]. Available: https://pytorch.org/docs/stable/generated/torch.nn.NLLLoss.html
[15] Z. Zhang, “Classical ml equations in latex.” [Online]. Available: https://blmoistawinde.github. io/ml_equations_latex/
[16] S. Vani and T. V. M. Rao, “An experimental approach towards the performance assessment of various optimizers on convolutional neural network,” in 2019 3rd International Conference on Trends in Electronics and Informatics (ICOEI), 2019, pp. 331–336.
[17] “Cs231n convolutional neural networks for visual recognition.” [Online]. Available: https://cs231n.github.io/neural-networks-3/
[18] J. Price, A. Wong, T. Yuan, J. Mathews, and T. Olorunniwo, “Stochastic gradient descent,” 2020. [Online]. Available: https://optimization.cbe.cornell.edu/index.php?title=Stochastic_ gradient_descent
[19] D. P. Kingma and J. Ba, “Adam: A method for stochastic optimization,” arXiv preprint arXiv:1412.6980, 2014.
[20] J. Huang, “Rmsprop,” 2020. [Online]. Available: https://optimization.cbe.cornell.edu/index. php?title=RMSProp
[21] A. Ajagekar, “Adam,” 2021. [Online]. Available: https://optimization.cbe.cornell.edu/index. php?title=Adam
[22] E. Landman, “difpy.” [Online]. Available: https://github.com/elisemercury/ Duplicate-Image-Finder
[23] G. Bradski, “The OpenCV Library,” Dr. Dobb’s Journal of Software Tools, 2000.
[24] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, “Scikit-learn: Machine learning in Python,” Journal of Machine Learning Research, vol. 12, pp. 2825–2830, 2011.
[25] T. P. L. t. Falcon William, “Pytorch lightning.”
[26] L. Biewald, “Experiment tracking with weights and biases,” 2020, software available from wandb.com. [Online]. Available: https://www.wandb.com/
[27] I. Kandel and M. Castelli, “The effect of batch size on the generalizability of the convolutional neural networks on a histopathology dataset,” ICT Express, vol. 6, no. 4, pp. 312–315, 2020. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2405959519303455 [28] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, “Dropout: A simple way to prevent neural networks from overfitting,” Journal of Machine Learning Research, vol. 15, no. 56, pp. 1929–1958, 2014. [Online]. Available: http://jmlr.org/papers/v15/srivastava14a.html
[29] W. Ahmed and A. Karim, “The impact of filter size and number of filters on classification accuracy in cnn,” 04 2020, pp. 88–93.
[30] Kloiti, Is Big Chungus the real official fanon version of Terence? Angry Birds Fanon wiki, 2021.