

Image Classification, Augmentation and Model Explanation with CNN, GAN, & Grad-CAM

Zhenkun Wang, Zhiyu Lin, Jianhao Ji

zw206@georgetown.edu, zl281@georgetown.edu, jj913@georgetown.edu

ABSTRACT

Different species of birds have very different appearances. In this project, we apply data augmentation with GAN, train a few neural networks including CNN, VGG16, ResNet and Xception to identify bird species given the image, present the prediction evaluation metrics (accuracy and confusion matrices), and explain the model prediction through a few model explainers including LIME, SHAP, and Grad-CAM.

INTRODUCTION

Image classification or computer vision is one of the hottest topics in deep learning lately. It is useful in a wide variety of fields, including healthcare, e-commerce, forensics, manufacturing, etc. For example, computer vision models could perform classification tasks on CT scans for cancer diagnosis, recommend products for online shopping websites, flag spam pictures, and detect defects for manufactured products. Computer vision models' importance, flexibility and robustness have been proven in numerous use cases over the past decade, so we decide to focus on computer vision for this project.

Instead of training an image classification model and being done with it, we are interested in the entire data processing pipeline: data augmentation, model training/evaluation, and prediction explanation.

Model explainability is gaining popularity because there is increasing demand for building explainable AI from legal and business points of views. Ras et al point out that explainable AI is important in "users, laws & regulations, explanations and algorithms"

RELATED WORK

Due to the time and hardware requirements to train good computer vision models, lots of work in this field has been dedicated to transfer learning. Some famous pre-trained computer vision models include VGG19, ResNet, Inception, Xception, etc. These models are largely trained on ImageNet with millions of parameters fine-tuned.

For data augmentation, there has been plenty of research around geometric transformations, color augmentations, kernel filters, image mixtures, random erasing, as well as adversarial training. One of the complex methods is training Generative Adversarial Networks (GANs) for image augmentation.

For model explainability, some relatively simple methods include LIME (Local Interpretable Model-Agnostic Explanations) and SHAP (SHapley Additive exPlanations), which build simple, explainable surrogate models to replicate model results. Other more DNN-specific methods include Activation Maximization (AM), Layer-wise relevance propagation (LRP), Knowledge Graphs (KG), tree-based hierarchical methods, and many more.

DATA

Our project takes the bird species dataset from Kaggle. This dataset contains images of 230 birds species, and each species has on average 140 images. Due to hardware and time limitations, we are only taking the top 10 species to perform our image classification task.



figure: Top 10 bird species from 230-bird-species dataset

METHODS

We are applying two methods for data augmentation: regular image stretching, reshaping, and transformations as well as image augmentation with GANs. In this case, we are using GANs to generate more bird images from the original training data to achieve a better modeling result. We compare this result with the model without any augmentation and present the performance increase.

For model training, we train a simple CNN for benchmarking, but also apply a few pretrained models such as Xception, VGG19.

Finally, we explain the 'black-box' models with some model explainers including LIME, SHAP, and a computer-vision specific approach with Grad-CAM (Gradient-weighted Class Activation Mapping). We also compare these three explanations.

DATA AUGMENTATION

1. CNN Data Augmentation

The CNN model contains 3 convolutional layers followed by max pooling and 20% dropout. This CNN model only performs basic data augmentation procedures, including upto 40% rotation, 0.2 width and height shift, up to 20% zooming as well as horizontal flipping.

2. GAN Data Augmentation

In our case, 5 Conv2DTranspose and 6 LeakyReLU layers were added into the generator and similar structures in discriminator. We used the african crowned crane as the training set and did 20,000 iterations which is about 9,000 epochs. However, since the image size is far larger than datasets like mnist, the result is not perfect enough, where only the basic color and structure is close to the real image



figure: CNN data augmentation



figure: GAN data augmentation

MODELS

1. CNN

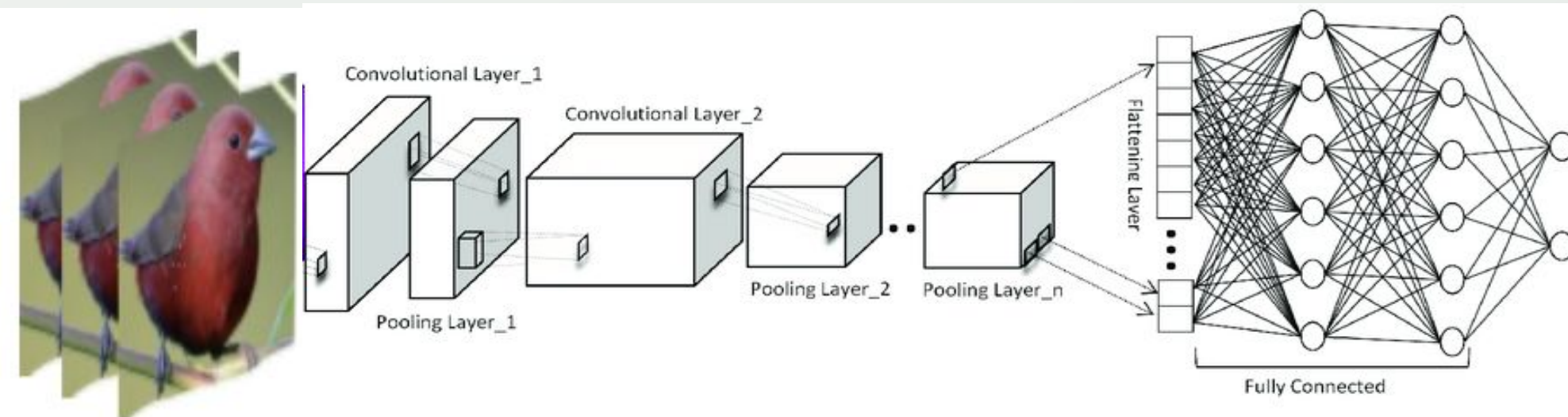
We first trained a CNN model. This model contains 3 convolutional 2d layers, each followed by a max pooling layer and a dropout layer to avoid overfitting. This model utilizes the Adam optimizer and was trained through 50 iterations. It was only able to achieve 76% classification accuracy on the test data.

2. VGG19

Besides CNN, we also tried the pre-trained model, VGG19, on our data. The VGG19 model contains 19 conv layers and 3 fully connected layers. This network on our dataset out-performed the CNN model and reached a 91% accuracy on the test data.

3. Xception

Xception has been trained on 350 million images with over 17,000 classes. This network on our dataset out-performed the CNN model and reached a 98% accuracy on the test data.



Model	Test Accuracy
CNN	0.76
VGG19	0.91
Xception	0.98

REFERENCE

1. Montavon, Grégoire, et al. "Methods for Interpreting and Understanding Deep Neural Networks." *Digital Signal Processing, Academic Press*, 24 Oct. 2017, www.sciencedirect.com/science/article/pii/S1051200417302385.
2. Daniels, Zachary A., Frank, Logan D., Menart, Christopher J., Raymer, Michael, and Hitzler, Pascal. "A Framework for Explainable Deep Neural Models Using External Knowledge Graphs," Proceedings Volume 11413, *Artificial Intelligence and Machine Learning for Multi-Domain Operations Applications*; 1141338 (2020). International Society for Optics and Photonics.
3. Marino, Kenneth, Salakhutdinov, Ruslan, and Gupta, Abhinav, "The more you know: Using knowledge graphs for image classification," *Computer Vision and Pattern Recognition*, 2673–2681, 2017.
4. Guo, Yanming, Liu, Yu, Bakker, Erwin M., Guo, Yuanhao, and Lew, Michael S., "Cnn-rnn: a large-scale hierarchical image classification framework," *Multimedia Tools and Applications*, 77(8), 10251–10271, 2018.
5. Leonardo, Matheus M., et al. "Deep Feature-Based Classifiers for Fruit Fly Identification (Diptera: Tephritidae)." *ResearchGate*, 31st SIBGRAPI Conference on Graphics, Patterns and Images (SIBGRAPI), 2018, www.researchgate.net/publication/330478807_Deep_Feature-Based_Classifiers_for_Fruit_Fly_Identification_Diptera_Tephritidae.
6. Shen, Xiaoyu. "Bird_Xception." *Kaggle*, 8 Dec. 2020, www.kaggle.com/xiaoyushen/bird-xception.
7. Chetoui, Mohamed. "Grad-CAM- Gradient-Weighted Class Activation Mapping." *Medium*, 29 Mar. 2019, medium.com/@mohamedchetoui/grad-cam-gradient-weighted-class-activation-mapping-ffd72742243a.
8. Ras, Gabrielle, van Gerven, Marcel, Haselager, Pim. "Explanation Methods in Deep Learning: Users, Values, Concerns and Challenges." In: Escalante H. et al. (eds) Explainable and Interpretable Models in Computer Vision and Machine Learning. The Springer Series on Challenges in Machine Learning. *Springer, Cham*. 2018
9. Brownlee, Jason. "How to Develop a GAN for Generating MNIST Handwritten Digits." *Machine Learning Mastery*, 1 Sept. 2020, machinelearningmastery.com/how-to-develop-a-generative-adversarial-network-for-an-mnist-handwritten-digits-from-scratch-in-keras/.

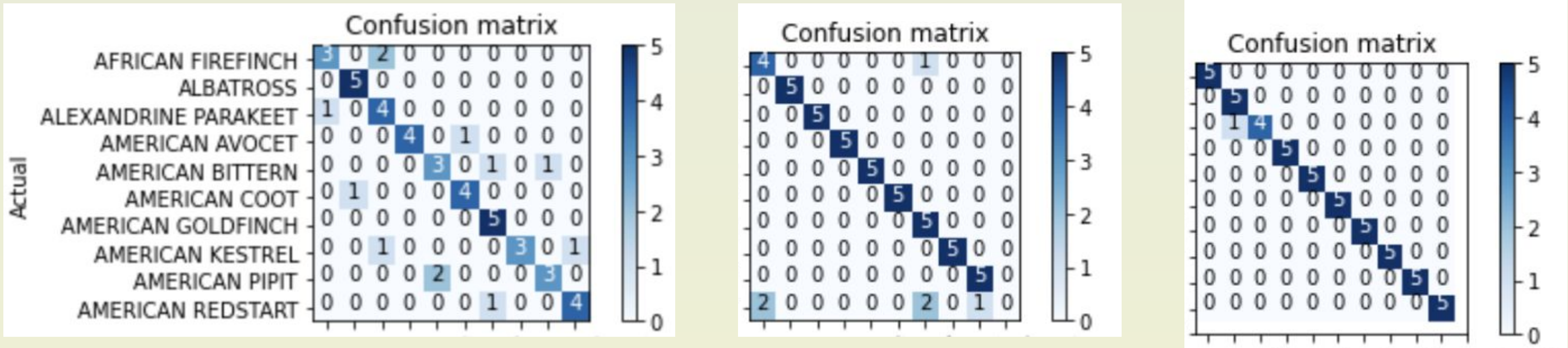


Figure 5. Test confusion matrices for CNN (left), VGG19 (mid) and Xception (right)

MODEL EXPLAINABILITY

1. LIME

From the illustration below, we see that LIME is not doing a great job of identifying the characteristics for the bird. The highlighted areas in the end are not distinguishing features for this type of bird.

2. SHAP

In our example, SHAP highlights the portrait outline, the red color block, and some backgrounds. SHAP correctly identifies the highest probability for the actual predicted class (top left), and it highlights the contour of the bird's head and the red color block.

However, this explanation is still not very satisfactory because we are expecting the explainer to "see" what we see as the unique characteristics of the bird, which would be centered around the body of the bird in this case.

3. Grad-CAM

Grad-CAM is developed from CAM. The CAM model is able to generate a heat map of the essential part in a picture. However, it has many restraints. To apply CAM on our model, we need to replace fully connected layers with global average pooling.

The Grad-CAM overcomes this problem. The Grad-CAM could produce multiple essential content layers without global average pooling. Therefore, we do not need to rebuild our model and get essential content heat maps directly.

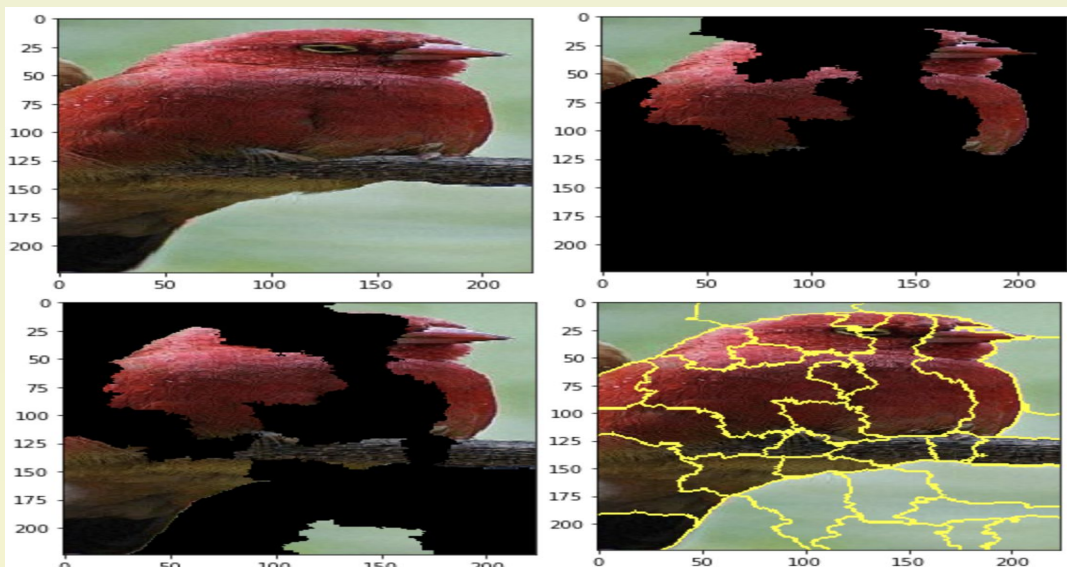


Figure: LIME explanation on the American Firefinch

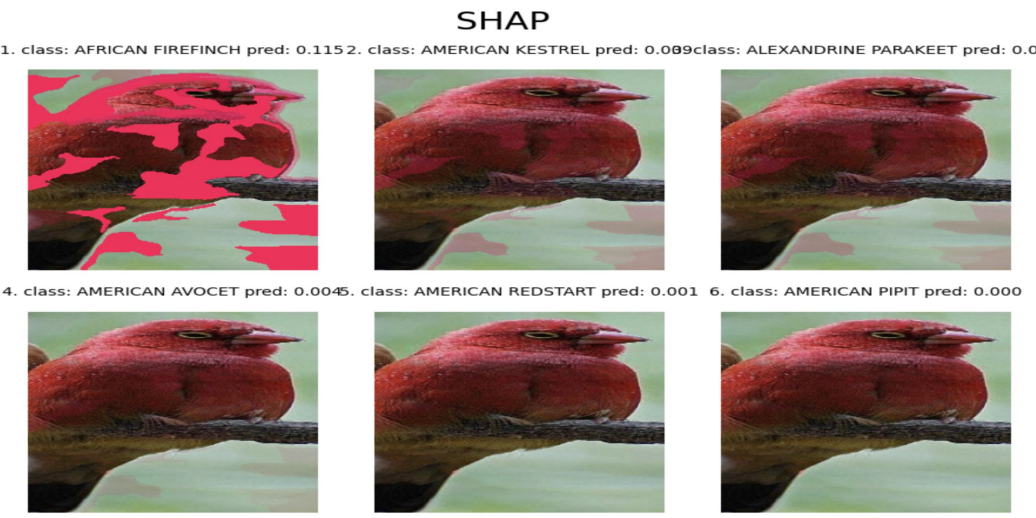


Figure: SHAP explanation of the American Firefinch



Figure: Grad-CAM explanation on the American Coot

CONCLUSION

In this project, we compared the performance of CNN models with and without data augmentation.

For data augmentation, we tried CNN ImageDataGenerator and GAN data augmentation. The first method only applies simple rotation, distortion, and flip to original pictures, while GAN adds noise data to original pictures. Due to the time limits, GAN only produced vague images. We would apply more training time to this process in the future.

For model performance, we found simple data augmentation dramatically boosts the model performance from 12% to 60%. Besides the CNN model, we also applied two pretrained models (VGG19 and Xception). In the end, we get 91% for VGG, and 98% for Xception.

In the model explainability section, we tried LIME, SHAP, and Grad-CAM methods. LIME and SHAP explain the model by determining each part of the picture, while the Grad-CAM only focuses on one essential part of the picture.

In the future, we might explore more model explainability methods for computer vision, and include more classes for training.

