Using Neural Networks to classify aircrafts in images

Shanu Vashishtha
UMass Amherst
svashishtha@umass.edu

Abstract

Aircraft models have wide variations between them. There are differences in their purpose, size, designation, structure, historical style, and branding. Navigation systems deployed at airports need a visual representation based classification system. The problem of fine-grained classification of these aircraft images is challenging because even though the models are visually distinguishable, the differences are subtle in many cases. Convolutional Neural Networks are widely adapted for performing the task of classification for different classes of objects. In this work, standard convolutional architectures have been explored to test their performance on classifying these images. We perform a simple feature extraction on these images followed by domain specific fine-tuning for different manufacturer and variant hierarchies of aircraft images. To improve upon the extracted features, we test the performance of bilinear CNNs on this particular classification task. Bilinear CNNs utilize existing feature extractors to combine local features before performing the classification. Our project demonstrates that these neural network architectures are most suited to the task of classifying aircraft images.

1. Introduction

Aircraft classification is a challenging task. We have a very wide range of aircrafts deployed all over the world for various purposes-commercial, freight, military etc. There are variations in models (Airbus A320 and A320neo have a difference of sharklets), variants(Boeing 737-200, 737-600 which differ in passenger capacities among other things), family(Boeing 737 and Boeing 787 differ in number of engines) and manufacturers(Boeing, Airbus, Bombardier etc). These variations have arisen because of historical reasons within the industry as well as the purposes which these aircrafts were supposed to fulfill. Recently, there has been a push for major airports to deploy camera based aircraft guidance systems. This necessitates the development of robust detection and classification systems for these aircrafts. A proper classification system can then provide guidance to that particular aircraft variant based on its properties. Hence, there is a need for a technology which can differentiate between aircrafts based on its visual properties.

In parallel to this requirement has been the development of neural networks as a classifier. Various architectures of neural networks have achieved remarkable performances on the Imagenet classification task which comprises of 1000 different objects. In comparison, aircrafts only have about 100 different variants for classification although the differences in between classes are fairly subtle in nature. In this work, we explore the performance of existing neural network architectures - Resnet, Alexnet, VGG16 and bilinear CNNs for this classification task.

The dataset used for the work is the FGVC-Aircraft Benchmark dataset[6]. It contains 10,200 images of aircraft, with 100 images for each of 102 different aircraft model variants. A few sample images from the dataset are shown in Fig.1.

The baseline for the task has been as described in the work here [6]. The baseline model is a non-linear SVM on a χ^2 kernel, bag-of-visual words, 600 k-means words dictionary, multi-scale dense SIFT features, and 1 x 1, 2 x 2 spatial pyramid with the model trained on the entire image instead of the bounding box information which is also provided along with the dataset. The overall accuracy reported for the classi-

fier is 48.69% measured using a confusion matrix approach.
To build upon the baseline, an existing classification network architecture was used as a feature-extractor for the images in this dataset. The performances were then compared after fine-tuning the network on the dataset. This was done for two hierarchies - Manufacturer and Variant as provided in the dataset. Based on the results obtained, experiments were carried out with bilinear CNN architectures for the classification task.

2. Related work

Alexnet[4] was perhaps the first implementation of neural networks to image classification task. Since then, different variants of convolutional neural networks have been applied to the task of image classification. Some of the variants proposed over the years are - VGGNet[7] and Resnets[2]. VGGNet of different layer depths have achieved significant results on the Imagenet classification task. The development of Various techniques like Batch Normalization and dropout [3] have been proposed to improve the performances of these networks and generalise them to other visual representation based tasks.
Applying these neural networks to the task of Fine-grained visual classification has been presented in [5]. These existing networks adapt robustly to the fine-grained classification task. However, with limited data availability, extracting additional combinations of features is hypothesised to improve the results. Such a task is achieved using Fisher vectors in the literature[1]. As CNNs are a natural feature extractor, the work of Lin et al[5] also proposes bilinear CNN architectures which leads to an improvement in the classification tasks for birds, cars and aircrafts. In this project, we follow a similar approach for the classification task which is focused on aircrafts.

3. Approach

In this project, VGGNet, AlexNet and Resnet were trained from scratch initially. The dataset used for the task is the FGVC-Aircraft Benchmark dataset. The (main) aircraft in each image is annotated with a tight bounding box and a hierarchical airplane model label. The data is divided into three equally-sized training, validation and test subsets. These are the hierarchies

Figure 1. Example images from the FGVC-Aircraft dataset

present in the dataset -

- **Model** - The variation in aircraft models(for example Airbus A320 and A320new) may not be visually measurable since the rigid-body dimensions are usually similar in this category. Aircraft manufacturers use these in their product improvement iterations.

- **Variant** - The dataset provides 100 variants. This class label is less specific compared to the previous hierarchy.

- **Family** - Family differences arise owing to aircrafts having different passenger capacities. This one is an intermediate visual classification problem lying in between Variant and Manufacturer hierarchies.

- **Manufacturer** - The dataset provides images from 30 different manufacturers. All the aircrafts manufactured by the same company fall under one category.

Initial training of standard architectures was done on the Manufacturer hierarchy. As the performace of the architectures was below the baseline for this task, transfer learning was explored for this domain. These networks were subsequently finetuned to gauge the improvement in their performance for the classification task. Once the best possible results were obtained after carefully tweaking the hyperparameters for each of the network architectures, the results as described later were obtained.

Next, classification was carried out for the Variant hierarchy. After following the previous methodology of first using the existing weights as feature extractors and then, comparing the performance to the one after fine-tuning the weights, the idea of Bilinear CNNs was explored for this task as it was hypothesized that significant improvements could be achieved over the results obtained. These novel architectures provided substantial benefits over the previous architectures. A pictorial depiction of the same is presented in Fig 2.
The Bilinear CNN model combines the features extracted from two extractors to obtain a bilinear vector before performing the classification task. For this project, the CNN models used were pre-trained on the ImageNet dataset. This is useful because the data available solely from the dataset is in-sufficient to train the network for an acceptable level of performance. This also has the benefit of allowing images of arbitrary size to be processed by the CNN. To aggregate the features from the two CNNs, sum-pooling is performed across the features. The resulting vector is l_2 normalized after passing through a signed square root step.
The input images were resized to 448x448 and the features were extracted using two networks before a bilinear-combination, sum-pooling and normalization. It performs significantly well for the variant category classification. The feature extractor used in the original architecture is VGGNet. After evaluating its performance as it is, we explored a variation of the model where Resnets were used as the feature extractors before the pooling task.

4. Experiment

4.1. Manufacturer hierarchy

Classification task involving a total of 30 classes. The images were resized to 224x224 before passing them as input to the network. The weights were trained using Stochastic Gradient Descent with a learning rate of 0.01 and a cross-entropy loss function. The performance of standard architectures are listed in Table 1.
Upon performing a feature-extraction task with these existing architectures, we obtain a performance below the baseline. Although marginally below the baseline, but the weights of the network are trained only for classification of the new 30 classes for the fully-connected

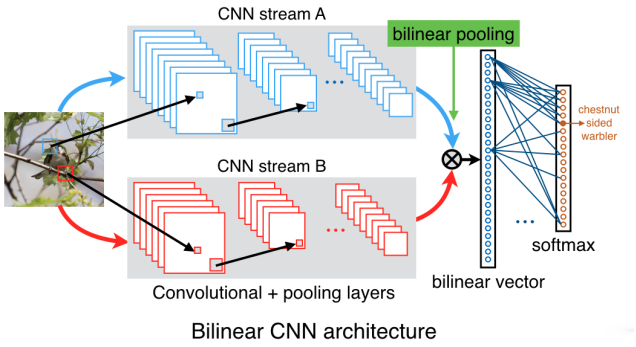

Figure 2. Bilinear CNN architecture

Network	Feature extraction	Fine-tuning
Resnet34	45.81%	78.09%
VGG11	45.84%	84.25%

Table 1. Accuracy results for different architectures on manufacturer hierarchy. Networks trained for a total of 60 epochs using SGD($\eta = 0.01, \epsilon = 10^{-4}$)

layers at this step. Fine-tuning the entire network on the dataset leads to an improvement in the classification task as noted in Table 1.
This change in the performance is observed to be higher in the case of VGGNet compared to the other models. A possible reason for the quantum improvement is that we have a lesser number of classes to train for at the current task as compared to the original ImageNet dataset which has 1000 different classes of objects. A key observation is that the training of Alexnet took a similar amount of time as that of the VGGnet. However, when compared to that of Resnet, the training time was about 3x lower even though the experiments were conducted using a 34-layer Resnet as compared to a 11 layer VGGnet with batch normalization.
We present samples of misclassified images for this task in figure 3. It can clearly be observed that the wrong predictions consist of images which are very challenging to distinguish visually. Even for the naked eye, images which consist of the aircraft's fuselage in a large portion of the image provide a tough classification task. Other instances included images where the fuselage merged with the background textures making it difficult to distinguish between the aircraft models. Also, our classification task was more accurate for classes with more image samples. Classes with fewer

Figure 3. Manufacturer images which were mis-classified by ResNet34

samples were more likely to have their predictions incorrect when evaluated on the test-set.

4.2. Variant hierarchy

Classification task involving a total of 100 classes. The performance of standard architectures are listed in Table 2. Similar training parameters were used as described for the manufacturer hierarchy. This task is a challenging one because of the subtle visual variations in between the models and we obtain a classification accuracy which is again below the baseline.

Proceeding as before, we perform finetuning of the architectures. The observed results are reported in Table 2. We observe that the VGGNet outperforms other architectures after finetuning the weights of the entire network. This could be because of the weights which are updated in case of VGGNet which are more than that of other architectures.

Comparison in terms of training time gives us a value which is three times lower for VGGNet vis-a-vis Resnet. Resnet-34 trained much faster compared to VGG-11 with batch normalization. Alexnet took a similar amount of time for training as compared to VGGNet. However, it had a poor accuracy on the classification task for the test set images.

We present a few of the mis-classified images for this category in figure 4. Even though we observe an increase in the performance of the classification task, the best performance observed at the fine-grained task seems low for a standard architecture. To improve the same, we explore the idea of Bilinear CNNs.

Network	Feature extraction	Fine-tuning
AlexNet	23.52%	32.04%
VGG11	49.26%	60.6%
ResNet34	31.08%	54.72%

Table 2. Accuracy results for different architectures on variant hierarchy. Networks were trained for a total of 60 epochs using SGD($\eta = 0.01, \epsilon = 10^{-4}$)

4.3. Bilinear CNNs

This architecture utilizes CNNs as feature extractors. It takes the features after the last pooling layers from 2 CNNs, computes their outer product and from the resulting matrix, performs a summation before utilizing the fc-layers for the k-way classification task. In the original paper [5], bilinear architecture has been described using the VGG-16 layers as the CNN. In this project, since we started off with VGG-11 architecture, we continue with that as our feature extractor for the bilinear model. This is to have a more consistent evaluation metric for the task at hand.

The first step here was again to finetune the classification layer for the task keeping the CNN pre-trained on the Imagenet weights. The next step involved finetuning the weights of the entire network as has been the procedure followed till now. The results for the trial runs are presented in Table 3. As expected, we observe a quantum jump when we fine-tuned the weights of the entire network on the current dataset. These numbers are also similar to that of manufacturer hierachy classification as reported previously. In the original work, 8% higher accuracy is reported on the test set. This could be due to the fact that our model used VGG11 architecture while theirs is based on VGG16 architecture.

Continuing with the exploration of reducing the training time, we tweaked the bilinear model to use a ResNet34 architecture in place of VGG11. The results for the same are reported in table 3. The performance accuracies obtained here are lower that that of VGGNet. This result is similar to that of the previous case with standard architectures. Another similarity was in the training time which was observed to be 2.5x lower in case of Resnet34 as compared to that of VGG11.

Network	Feature extraction	Fine-tuning
VGG11	67.3%	76.5%
ResNet34	56.6%	70.2%

Table 3. Test-set accuracy results in case of Bilinear models. Networks were trained for a total of 100 epochs using SGD($\eta = 0.05, \epsilon = 10^{-5}$)

Figure 4. Variant images which were mis-classified by ResNet34

5. Conclusion

In this work, we wanted to explore the task of classifying aircraft images using neural networks. For that process, we utilized an existing dataset namely FGVC-Aircraft. The performance of stand-alone architectures on these images weren't good. However, these CNNs are robust to domain adaptation. Specifically, to the task of fine-grained classification. The existing architectures responded differently to the task of network fine-tuning. VGGNet clearly outperformed other standard architectures like Resnet and Alexnet. An important observation in this experiment was that stacking more number of layers didn't necessarily improve the performance of the network as is observed in case of Resnet and VGGNet results.

Adapting specifically to the task of fine-grained classification, a local feature extractor performs better. This is because a local feature extractor can learn to differentiate between subtle variations between images. An aircraft for example can be differentiated visually by the fuselage length, its wingspan, the number of windows on its body. These are features which a local feature extractor is able to extract and hence, its performance at the fine-level classification task is better when compared to networks that do not look at this methodology of analysis.

A possible way to implement these local features is by using a bilinear CNN and their performance is quite good. The bilinear CNNs combine features by performing a pooling over the extracted features and this method guarantees good results for the classification task.

Analysing the mis-classified images in both the hierarchies across architectures, we observe that a high number of them belong to the set where only the fuselage occupies a large portion of the image. There are pictures where the aircraft is being viewed from below or the other end. These viewpoints make the classification task difficult. A possible way to be able to distinguish between them in a more robust manner will be to increase the training samples for minor categories. In the current dataset for the manufacturer variant, we observe that there are categories with only 30 images. Such a constraint makes it difficult to learn the features and hence, inaccuracies creep in the final evaluation. An increase in those samples should lead to an increase in the accuracy of the network.

5.1. Future Explorations

The following unexplored ideas came up during the course of the project and will be explored later on -

- End-to-end training of the entire bilinear model with the Resnet - VGGNet when trained end-to-end provided for the bilinear model provided an increase of 7% in accuracy results as reported in the work of [5]. A similar exercise when carried out with deeper Resnet architectures should lead to an increase in the classification performance.

- Combining hierarchical classification tasks in an efficient manner - Since a lot of features are similar across manufacturers when it comes to aircrafts, a possible classification task utilising this information should help in the overall task. This is based on the hypothesis that the aircraft manufacturers utilise components that are similar across their models and these components have similar visual features. The difference arises in the positioning of these components across the aircraft body. Thus, an effective classification architecture can utilise these features.

- Visualizing the features learned during the training to infer distinguishing aircraft components which the network learns - This will be extremely

helpful in analysing the mis-classified images. A possible idea to explore is when we are trying to perform a binary classification task in between two models. Such a visualising feature will help in knowing the root cause of mis-classification and hence, generalizing it later on to the multi-classification task.

- Exploring the idea of a trilinear model architecture - Trilinear architectures can help bring in additional features to the model. This could be based on optical flow, depth estimates or some gaussian filters. The idea is to capture variations which can help in the task of classification.

References

[1] M. Cimpoi, S. Maji, I. Kokkinos, and A. Vedaldi. Deep filter banks for texture recognition, description, and segmentation. *CoRR*, abs/1507.02620, 2015.

[2] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. *arXiv preprint arXiv:1512.03385*, 2015.

[3] S. Ioffe and C. Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *CoRR*, abs/1502.03167, 2015.

[4] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 25*, pages 1097–1105. Curran Associates, Inc., 2012.

[5] T.-Y. Lin, A. RoyChowdhury, and S. Maji. Bilinear cnn models for fine-grained visual recognition. In *International Conference on Computer Vision (ICCV)*, 2015.

[6] S. Maji, J. Kannala, E. Rahtu, M. Blaschko, and A. Vedaldi. Fine-grained visual classification of aircraft. Technical report, 2013.

[7] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *CoRR*, abs/1409.1556, 2014.