APS360 Project Progress Report: Commercial Aircraft Model Classification

Joey Yizhi Li
Student# 1010877044
joeyyizhi.li@mail.utoronto.ca

Ziyu Chen
Student# 1009857488
ziyujacob.chen@mail.utoronto.ca

Jici Ye
Student# 1009403879
jici.ye@mail.utoronto.ca

Deren Zhang
Student# 1008828440
deren.zhang@mail.utoronto.ca

1 Introduction

In this project, we propose a deep learning-based image classification system capable of recognizing commercial aircraft models from photographs. Accurately identifying commercial aircraft models from images has valuable applications in air traffic management, aviation analytics, and defense surveillance. However, manual identification of aircraft models from images is a demanding task, typically requiring extensive domain expertise. Differentiating between models often hinges on nuanced features such as slight differences in engine nacelle placement, wing sweep angle, landing gear configuration, or tail structure— details that are easily overlooked by the untrained eye. Thus, a machine learning model is the most effective approach to this problem.

Figure 1: Boeing 737-500, adapted from Maji et al. (2013c)

We build upon the FGVC-Aircraft dataset Maji et al. (2013a) by incorporating additional, up-to-date images of modern aircraft models and performing extensive data cleaning to ensure high-quality inputs. A sample input is shown in Figure 1, and we want the model to output its corresponding variant type. At the current stage, we choose a traditional Bag-of-Visual-Words (BoVW) model with a Support Vector Classifier (SVC) and an ANN classifier as the baseline. Our primary model explores state-of-the-art convolutional neural network (CNN) architectures, namely VGG-19 and ResNet-50, both modified with a Spatial Pyramid Pooling (SPP) layer to accommodate images of varying sizes without the need for explicit resizing. This work aims to assess the effectiveness of modern CNNs against classical methods in fine-grained, multi-class aircraft classification and to identify model architectures that generalize well under diverse viewing conditions. Our results will inform future efforts in applying AI for aviation-related visual recognition tasks.

2 INDIVIDUAL CONTRIBUTIONS AND RESPONSIBILITIES

Member	Responsibility	Progress
Joey	Developing baseline models	I have successfully implemented the baseline BoVW+SVC suggested by the author of the dataset using Python Maji et al. (2013b). I also modified this model by replacing SVC with a two-layer ANN classifier. The two baselines can now run on a small subset of the dataset. I also help to coordinate responsibility distribution and choose the model architecture.
Deren	Data preprocessing	I implemented automated scripts to detect and remove corrupted or duplicate images. I also conducted a manual inspection to exclude partial aircraft images. Finally, I split the cleaned dataset into training, validation, and test sets.
Jici Ye	Primary model implementation	I implemented the primary deep learning models for our project, including VGG19 and ResNet18 backbones with a Spatial Pyramid Pooling (SPP) layer and a two-layer MLP classifier. I trained both models for 30 epochs, analyzed training and validation accuracy and loss, and generated confusion matrices for evaluation.
Ziyu Chen	Data collection & processing	I built the new datasets and collected in total of 1500 new images for the newly added models to the original dataset and a final testing dataset using Jetphotos (2002). I also conducted a manual inspection to remove damaged, duplicate images, and exclude the images that only contain partial aircraft. In addition, I also helped to clean images and image augmentation.

2.1 TEAM COMMUNICATION

To ensure effective collaboration, our team follows a structured communication routine:

- **Weekly Zoom Meetings**: We hold regular meetings to discuss progress, address challenges, and align on upcoming tasks.
- **Instant Support via WeChat**: For quick questions or technical issues, we use WeChat to share insights and troubleshoot in real time.
- **Collaborative Coding on Google Colab**: All code development and testing are conducted on Google Colab, enabling seamless teamwork with shared notebooks and version control.

2.2 PROGRESS ASSESSMENT

- **On Track**: Baseline and primary model construction. The models can successfully compile and run on a small dataset.
- **Slight Delay**: Data collection behind scheduling. We still lack data for 5 airplane variants.
- **Risk Redundancies**:
  - Two members are now responsible for collecting data from Jetphotos (2002).
  - Backup GPU resources are identified if the primary hardware fails. We all have access to Colab Pro and A100 GPU. We also have an offline workstation equipped with an Nvidia 4070 GPU available as a backup.

2.3 Updated Project Plan and Work Distribution

Task	Responder	Internal DDL
Data Collection and Processing	Deren, Jacob	7/18
Result Collection on the full dataset	Joey, Jici	7/31
Human Baseline	Jacob	7/31
Illustration/Figures	All	8/5
Project Presentation		
Introduction & Background	Deren	8/10
Data Processing	Jacob	8/10
Baseline Model	Joey	8/10
Primary Model	Jici	8/10
Video Recording & Editing	All	8/12
Final Report		
Introduction & Background	Deren	8/11
Data Processing	Jacob	8/11
Baseline Model	Joey	8/11
Primary Model	Jici	8/11
Discussion	Deren & Jacob	8/13
Ethical Consideration	Joey	8/13

3 Data Processing

3.1 Data Collection

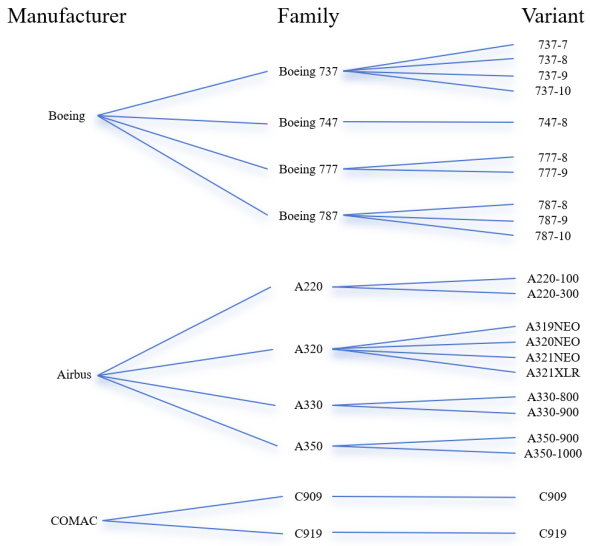

Figure 2: This graph shows the list of airplane models that are missing from the dataset. We will manually add images to update the dataset.

During the project proposal, we considered the newer models that we want to add to our dataset, as shown in Figure 2. After considering the number of images that can be found online, we decided to rule out A330-800 and A319NEO as both of them have fewer than 20 aircrafts that are currently operated by airlines. Regarding the Boeing 737-7, 737-10, 777-8X, 777-9X, they have not been in commercial use, which means they have a limited number of liveries and images, so they are ruled out, too. For A321XLR, it has the same appearance as the so it is difficult to identify the difference between them. In addition, it has only delivered 4 aircraft, so similarly, we decided to rule them out. We manually collected the images of the 15 aircraft models, each with 100 images. Our newly collected data is obtained from *Jetphotos.com* Jetphotos (2002).

Aircraft Name	Search Name in JetPhotos
A220-100	Airbus A220-171
A220-300	Airbus A220-371
A320NEO	Airbus A320-251/271N
A321NEO	Airbus A321-251/271NX
A330-900	Airbus A330-941
A350-900	Airbus A350-941
A350-1000	Airbus A350-1041
B737-8	Boeing 737-8 Max
B737-9	Boeing 737-9 Max
B747-8	Boeing 747-89L/830/8B5
B787-8	Boeing 787-8 Dreamliner
B787-9	Boeing 787-9 Dreamliner
B787-10	Boeing 787-10 Dreamliner
C909	COMAC C909/ARJ21-700
C919	COMAC C919

Figure 3: The figure shows the list of names that were used to search the specific aircraft in Jetphotos (2002). The difference between the search name and the Aircraft name for Airbus products comes from the engine selection. For A320NEO and A321NEO, '-251' represents the PW engine and '-271' represents the CFM engine. For Boeing 747-8, the difference comes from the suffix named by the operator. They represent Air China, Lufthansa, and Korean Air, respectively. For COMAC C909, ARJ21-700 and C909 are the same thing as COMAC officially renamed the aircraft on November 12th, 2024, at the China International Aviation & Aerospace Exhibition.

For image collections, we follow the following criteria:

1. Different Airline Liveries
2. Different Camera
3. View of the full aircraft body

According to these rules, when collecting images for best-selling models such as B737-8, A320NEO, A321NEO, etc, we mainly focused on choosing different liveries and angles for them. However, for A220-100, B747-8, and other low-sales models, we mainly focused on choosing images that have different angles, environments, and lights. After choosing suitable images, we downloaded them into their sets and named them from 1-100 to keep track of the number of images that have already been collected. All images are in the form of *.jpg* and they are uploaded to the shared Google Drive folder for further actions.

Regarding the final testing data, we will collect new images that have never been used in the original or updated dataset. For each aircraft model, we will collect 15 new images on Jetphotos (2002) for the final testing, and those images will not be used in any steps before the final test.

3.2 Data Inspection & Cleaning

```
Checking: 220-100
Checking: 220-300
Checking: 330-900
Duplicate: /content/drive/MyDrive/Colab Notebooks/Newly added images/330-900/23.jpg is a copy of /content/drive/MyDrive/Colab Notebooks/Newly added images/330-900/6.jpg
Checking: 350-900
Checking: 737-9
Checking: 747-8
Checking: 350-1000
Checking: 787-8
```

Figure 4: Output of the automated script showing that a duplicate has been found in A330-900 on pictures 6 and 23

Since we have already completed 10 classes of airplane model collections with 100 images per category, we implemented automated inspection scripts to detect corrupted or unreadable images and to identify duplicate files. Duplicate images were found in the A330-900 model category as shown in Fig. 4 and were manually removed and a new image was added instead.

As stated in the project proposal, the use of Spatial Pyramid Pooling (SPP) eliminates the need for image resizing or enforcing uniform resolution. However, our criteria indicates that each image should capture the entire airplane, as many aircraft models have similar features when only parts (such as engines or wings) are visible, and partial views can be misleading. To ensure this, all images were manually reviewed, and those showing only partial aircraft were excluded. We then manually add new images to replace the removed images so that each aircraft model would have 100 images.

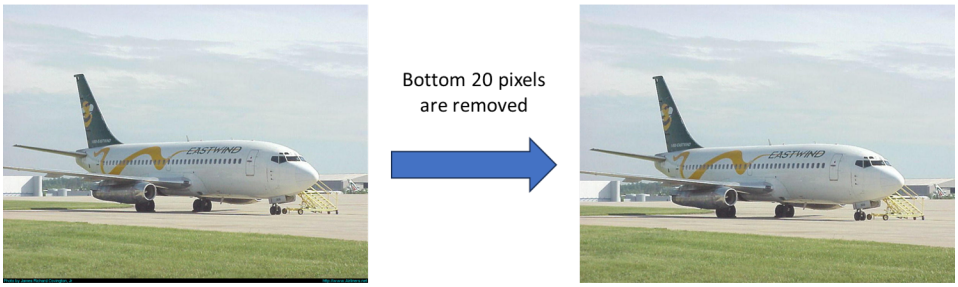

Figure 5: The bottom 20-pixel banner that includes copyright information is removed.

Additionally, the images in the original FGVC dataset contain a 20-pixel banner at the bottom for copyright. We remove them as the authors suggest Maji et al. (2013c), as shown in Fig 5. In addition, the watermarks on the images were also be removed.

With the dataset now cleaned and ensuring the number of images for each set is the same, the remaining images were randomly split into training, validation, and test sets using a 70:15:15 ratio.

3.3 Data Augmentation

Figure 6: Examples of data augmentation including horizontal flip, color jitter, and blur applied to the original image

Data augmentation was implemented through automated code after the dataset is finalized and all aircraft variants are added. Techniques such as random rotations, horizontal flips, color jittering, and blurring will be applied as shown in Fig. 6. These augmentations help the model learn features that are invariant to changes in pose, lighting, or orientation, thereby reducing overfitting. Augmentation will be applied only to the training set.

4 Baseline Model

The authors of the dataset used an SVC model as a baseline Maji et al. (2013b). Specifically, they selected a non-linear SVM on a χ^2 kernel, bag-of-visual words (BoVW), 600 k-means words dictionary, multi-scale dense SIFT features, and 1×1, 2×2 spatial pyramid. Essentially, the model first extracts local texture patterns like the number of engines, the shape of wings, and more; then counts their occurrence (as visual words); notes their rough spatial location via a spatial pyramid; and lastly uses an SVM to learn patterns in these histograms to separate airplane classes. An example

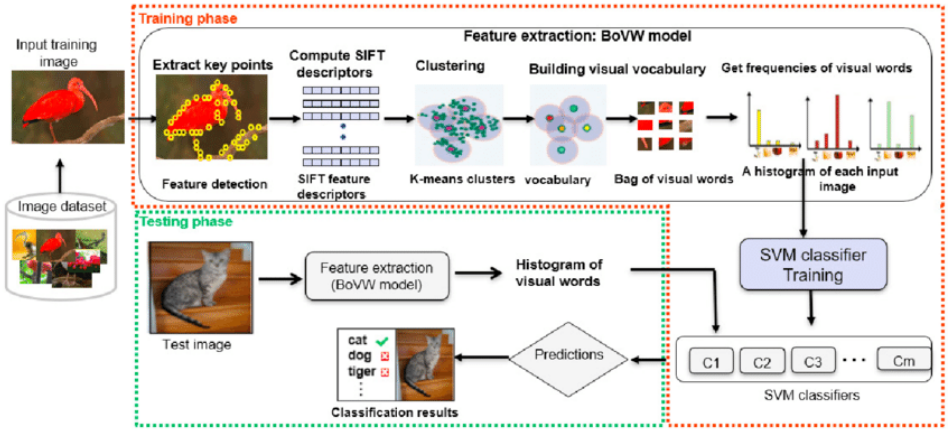

Figure 7: An Illustration of the BoVW+SVC Approach in Image Classification Task. Adapted from Filali & Zghal (2021)

pipeline is shown in Fig 7. Since this is the baseline the authors of the dataset used, we chose this approach as our baseline model.

However, since this paper was published in 2013, this traditional approach does not leverage the advancements and power of deep learning networks. To improve performance, we modify the pipeline by replacing the SVM classifier with a MLP in the final classification step.

4.1 Result

Since we are only focusing on major commercial aircraft, we choose aircraft from Boeing and Airbus from the FGVC dataset. In total, there are 34 variants, each with 100 images. Then, we split the dataset into a 70-15-15 train-validation-testing partition. The images are converted to greyscale as it is a standard SIFT procedure. We first randomly sample 150 images from the training set and extract their feature through SIFT to obtain the K-means cluster model. Then, the BoVW features of each image are extracted through our pipeline. The features from each image are a 3000-long `NumPy` array since the spatial pyramid will produce 5 bins, and each bin contributes to 600 features. Then, the features are stored and sent to the classifier.

4.1.1 SVC

Since all of the hyperparameters of this architecture are fixed, there is no need to validate the architecture. Thus, we use the training set to train the SVC, then use the model to fit the testing set to obtain the final performance.

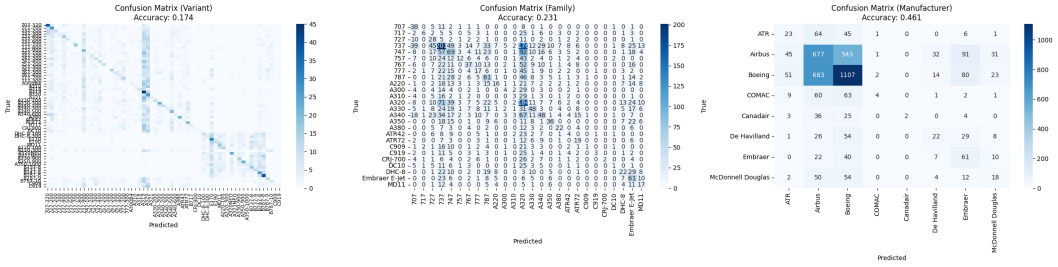

Figure 8: SVC on the Training Set. Boeing is manufacturer 0, and Airbus is 1. An interesting observation is that many aircraft are predicted as variant 24 and 25, which correspond to A319 and A320, two A320 variants from the family 9.

SVC performs very badly on both the training set and the testing set, reaching an accuracy of 17.4% and 5.9%. This is just slightly better than guessing. But this is reasonable given that the small

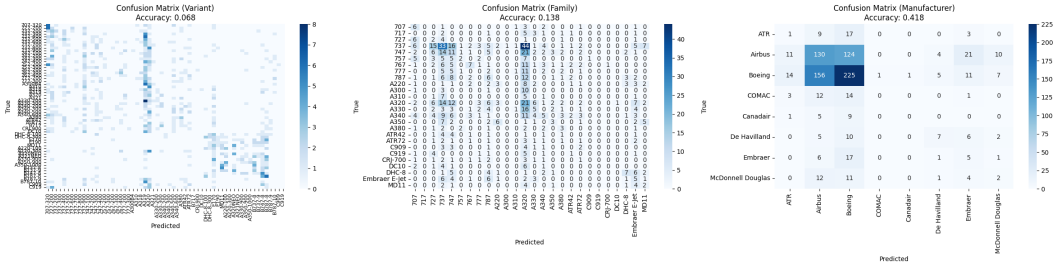

Figure 9: SVC on the Testing Set. SVC's performance between the two sets are quite similar: both have many predicted as A319 and A320.

variation between variants under the same model is extremely hard to identify. Additionally, we only choose Airbus and Boeing aircraft in the dataset, so they all look similar. These results are also reported by the author of the dataset Maji et al. (2013c).

4.1.2 MLP

BoVW is implemented using `sklearn`, so its output is a `NumPy` array. However, `Pytorch` uses tensor, so one of the challenges is to build a new `BoVWDataset` that inherits from `Dataset` of `torch.utils.data`. We implement a three-layer ANN classifier, where the input size is 3000, the first layer size is 1028, the second layer size is 256, and the output size is 34. ReLU is used as the activation function. We use `Adam` as the optimizer and `CrossEntropyLoss` as the loss function. The model is trained over 50 epochs with a batch size of 64 and a learning rate of 0.003.

The loss and error curve in Fig 10 shows that the losses of the training set keep decreasing, despite the losses of the validation set reaching a plateau at around the Epoch of 25. This demonstrates the MLP's ineffectiveness in generalizing the information of the training set.

(a) Loss over Epoch

(b) Error over Epoch

Figure 10: Training performance metrics

Likewise, MLP only performs a bit better than SVC with a higher testing accuracy of 13.9%, as shown in Fig 11. Since both SVC and MLP perform badly, we can confirm that the SIFT+BoVW approach can not extract important features from the images effectively, leading to bad classification results.

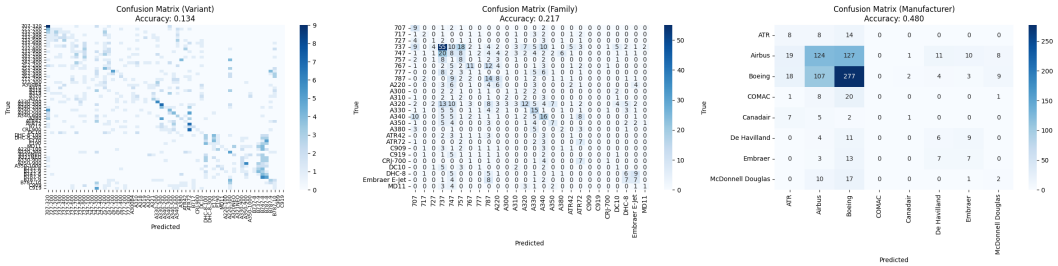

Figure 11: MLP's performance on the Testing Set. Performance is better than SVC, but still is undesirable.

4.2 CHALLENGE

- `OpenCV2` does not support GPU acceleration, and the BoVW features of the 3400 images took 4 hours. Thus, image augmentation has not been used so far. We will address this problem by running the BoVW feature extraction concurrently on all of the team members' colab terminals.
- The baseline models perform very badly. Although the result is reasonable, we hope to establish a stronger baseline for comparison with our primary model. Thus, we plan to use a human baseline. Our team member Jacob is an expert on commercial aircraft models. He will manually identify the aircraft models using the same testing set as our models.
- The MLP model is very prone to overfitting. Using regularization techniques like normalization may address like problem. Augmentation may help the model to perform better.

5 PRIMARY MODEL: VGG19 AND RESNET18 WITH SPP AND MLP CLASSIFIER

We implemented two primary models for fine-grained aircraft classification:

- **VGG19** + Spatial Pyramid Pooling (SPP) + MLP
- **ResNet18** + Spatial Pyramid Pooling (SPP) + MLP

Both models are based on pretrained convolutional backbones from ImageNet and use a two-layer MLP classifier. The SPP module performs multi-level spatial pooling with bins $[1 \times 1, 2 \times 2, 4 \times 4]$, resulting in 21 pooled regions per channel. Flattening the 512-channel output results in a $10{,}752$-dimensional feature vector, which is passed to a two-layer classifier. The first layer has 1024 neurons, and the output layer has 100 neurons. ReLU and dropout are used in between. Training was performed for 30 epochs with batch size 32, learning rate 1×10^{-4}, the Adam optimizer, and Cross Entropy Loss. To compare the primary model with our baseline, we use the same set of data with 34 variants obtained from Maji et al. (2013c)

WHY THIS ARCHITECTURE MAKES SENSE

- Pretrained CNNs (VGG19, ResNet18) provide robust, transferable low- and mid-level features.
- SPP introduces spatial robustness and scale invariance, enabling variable-sized inputs. This will prevent loss of important features due to image resizing, especially in our task, where aircraft variants may differ by only the shape of the wings.
- The MLP classifier is simple, fast, and effective for multi-class classification.
- The entire pipeline is modular and easy to reproduce using PyTorch and torchvision.

5.1 RESULTS

Figures 12 and 13 show the training and validation accuracy over 30 epochs. VGG19 demonstrates faster convergence and better generalization, while ResNet18 shows slightly lower performance overall. The results of both models are reported in Table 3.

Table 3: Training and validation accuracy comparison of VGG19 and ResNet18 with SPP.

Model	**Best Train Accuracy**	**Best Validation Accuracy**
VGG19 + SPP	98.2% (Epoch 30)	43.7% (Epochs 22–30)
ResNet18 + SPP	83.1% (Epoch 30)	39.4% (Epoch 28)

We applied the two model to the testing set, and their respective accuracy is 42.4% and 40.2%. This is much better than the performance of the baseline model, meaning the pretrained convolutional layers can better extract important features from the images than the SIFT+BoVW approach does. However, the result is still unsatisfactory. Since the current dataset we used has not undergone augmentation due to time and resource limitations, we believe that with more augmented data,

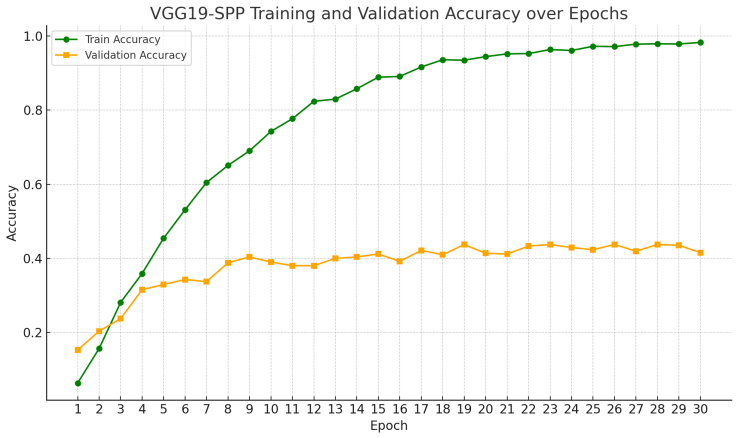

Figure 12: The accuracy curve of VGG19 over epoch

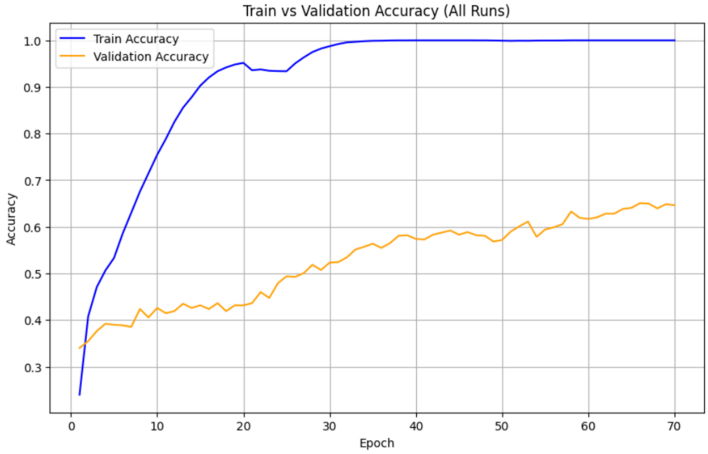

Figure 13: The accuracy curve of ResNet18

our model can better generalize the features of different aircraft models, leading to higher testing accuracy.

Challenges Faced

- PyTorch failed to download pretrained weights locally due to SSL errors. We manually downloaded the weights and placed them in the cache directory to resolve this.
- Training on Mac without GPU was extremely slow. We migrated to Google Colab with free GPU access, achieving over 10× training speed improvement.
- Ultimately, we will choose between the two pretrained CNN backbones. We will try to optimize the performance of both background as much as we can, and objectively select the one with the best performance in the future.

REFERENCES

Jalila Filali and Hajer Zghal. Comparing hmax and bovw models for large-scale image classification. *Procedia Computer Science*, 192:1141–1151, 01 2021. doi: 10.1016/j.procs.2021.08.117.

Jetphotos. Aviation photos - 5 million+ on jetphotos, 11 2002. URL https://www.jetphotos.com/.

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

Subhransu Maji, Esa Rahtu, Juho Kannala, Matthew Blaschko, and Andrea Vedaldi. Fine-grained visual classification of aircraft, 2013b. URL https://arxiv.org/abs/1306.5151.

Subhransu Maji, Esa Rahtu, Juho Kannala, Matthew Blaschko, and Andrea Vedaldi. Fine-grained visual classification of aircraft, 06 2013c.