Predicting planes in Microsoft Flight Simulator

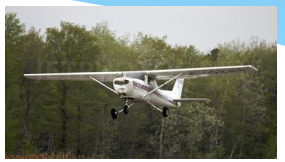
Oliver Bollverk, Richard Nõmm, Ludvig Leis, Jon Hendrik Aruväli

Github: https://github.com/o-bollverk/msf

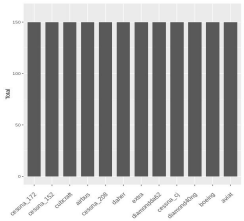
The goal and the problem

* Combining data from MSF and jetphotos.com
* Goal: to classify plane types
* A way to simulate plane type classification in the real world as MSF graphics are very realistic (air traffic control)

Process graph

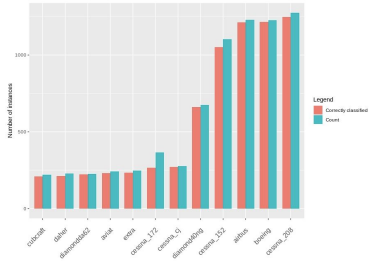


Balanced data (subset)



random selection

Unbalanced data



Train/test split

- Image augmentation → Resnet18
- Custom CNN → resize (64x64)
- Resnet34
- Resnet50

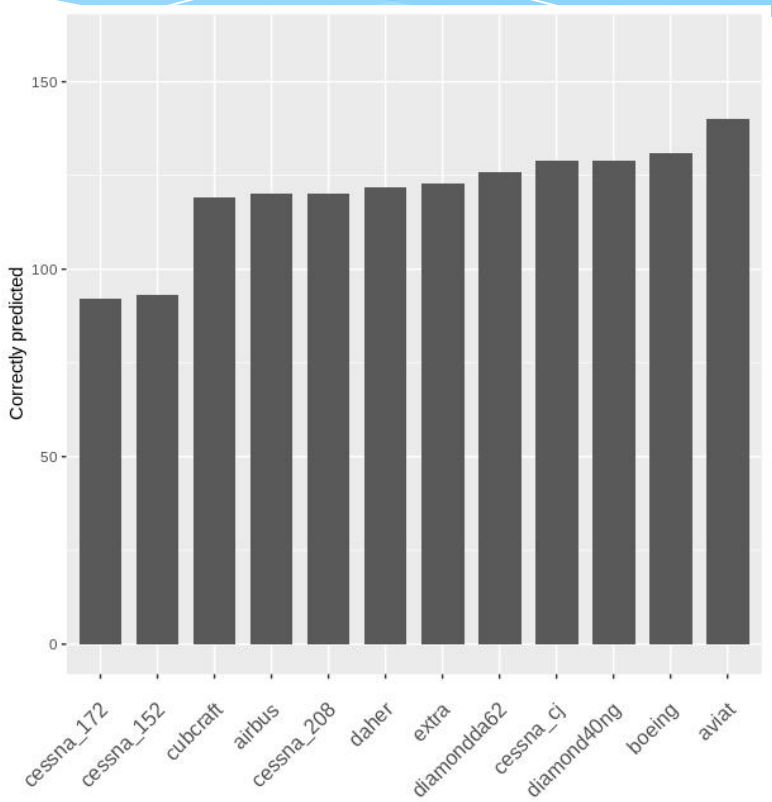
→ Accuracy scores (train accuracy and loss + test accuracy)

Train/test split

- Resnet34 → Accuracy (96%)
- Custom CNN → resize (64x64) → Accuracy (79%)

Methods/models comparison (balanced subset)

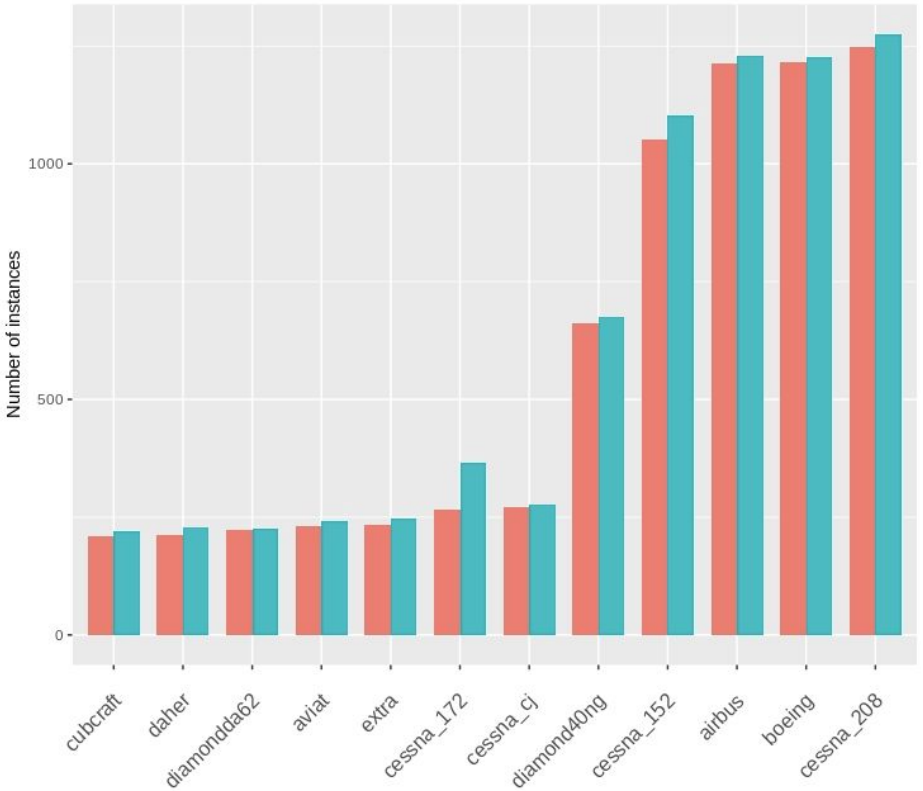
Model	Augmentation	Top test accuracy (with 15 epochs, balanced data) 3.5k images (80/20 ratio)
Resnet50	-	80%
Resnet34	-	76%
Resnet18	-	69%
Resnet18	Background elimination	69%
Resnet18	Mean filtering	69%
Resnet18	Dense image wrap	59%
Resnet18	Random hsv (color change)	70%
Resnet18	Rotation	68%
Custom CNN*	Image resize 64x64	66%
Custom CNN*	Image resize 64x64, mean filtering, wrap transform	65%



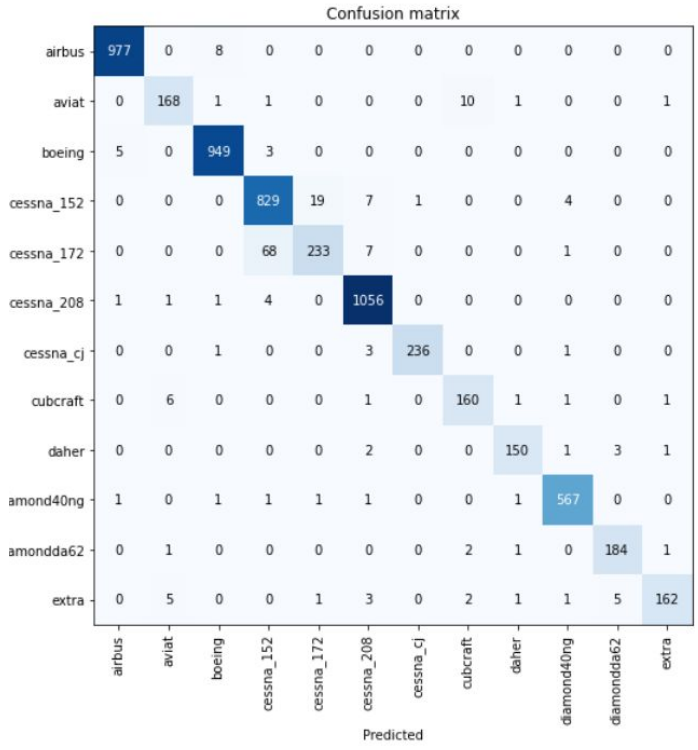
* Custom CNN used more epochs (128).

Results on unbalanced set

Resnet34

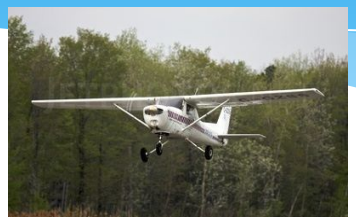


Resnet34

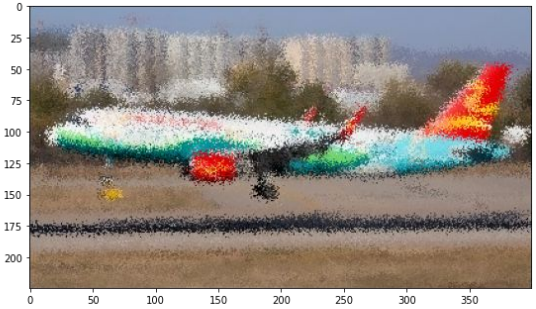


Resnet18 test with unbalanced data (5 epochs) - 29k images (80/20 ratio) - **94**% accuracy on test set.
Resnet34 test with unbalanced data (5 epochs) - 29k images (80/20 ratio) - **96**% accuracy on test set.
Custom CNN (Image resize 64x64) with unbalanced data - 29k images (80/20 ratio) - **80**% accuracy on test set.

Image augmentation techniques



Mean filtering



Dense image wrap

Random HSV in YIQ

Rotation

Expectations vs results

Expectations:

* 90+ percent accuracy
* Image augmentation will improve accuracy
* Background elimination will increase accuracy 5%
* Custom CNN accuracy 80%-90%

Results:

* Some augmentations decreased accuracy (wrap, grayscaling)
* Mean transform, background elimination and random hsv didn't change accuracy
* Higher resnets performed better
* Difference in accuracy on models trained on entire unbalanced dataset vs balanced subset was 70% to 94% (resnet18)
* Custom CNN performed worse than resnet18 on subsets (best ca 80% accuracy)

Lessons learned

* Competing with Resnet is difficult with custom CNNs
* Image augmentation technique selection is complicated and some methods may easily decrease accuracy (edge fading, grayscaling)
* Test and training set should be from the same distributions
* Online data collection challenges
* FastAI library is not to be recommended for building custom CNNs
* Image resize can be crucial to model efficiency
* Performance issues with Fastai (correct tensorflow versions, GPU drivers, etc)
* Personal PC > Google Colab

Thank you!