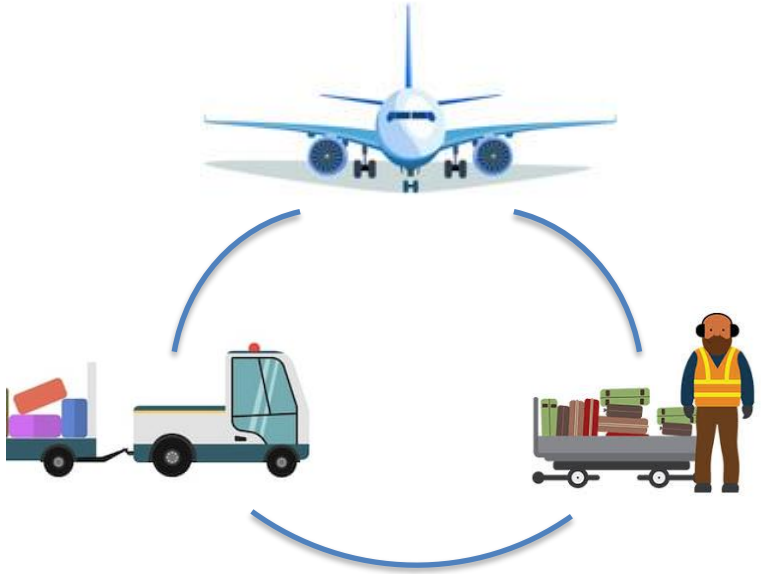
Final Project

Statistical Learning for DS
East Section

Ryan Cox, Casey Donnelly, Hunter Kempf, Olivia Lin

1,700+

Incidents per year



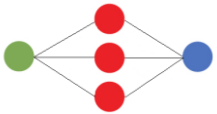
Goal: effectively reduce automobile and aircraft incidents at airports through real-time traffic management that leverages autonomous image detection and classification

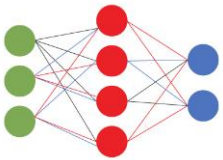
Source: FAA, 2019

Framework

Leverage available dataset → Explore and prepare data → Train base model → Optimize model → Draw conclusions

CIFAR-10
60,000 32x32 color images

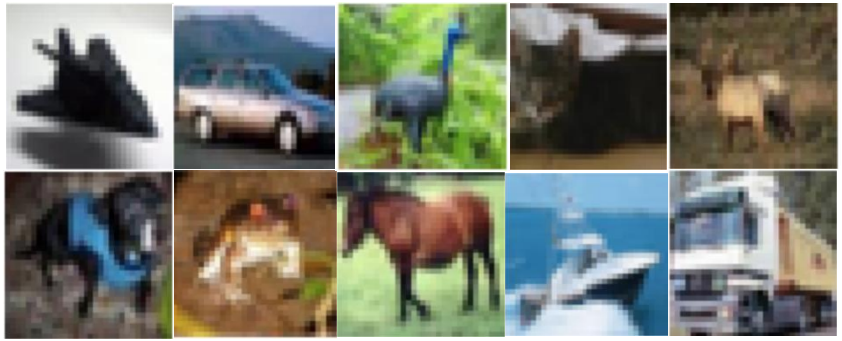




Exploratory Data Analysis

Data set designed for computer vision: size, number, complexity

- Data already split into training (50k) and test (10k) sets, further split the train set to create a validation set
- Data has numerous shortcomings in terms of quality, colors, and confounding objects



10 Unique Image Classes

- 0 → airplane
- 1 → automobile
- 2 → bird
- 3 → cat
- 4 → deer
- 5 → dog
- 6 → frog
- 7 → horse
- 8 → ship
- 9 → truck

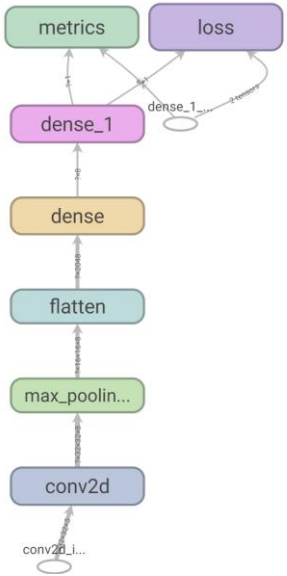
Exploratory Data Analysis

Can a computer beat manual human classification?

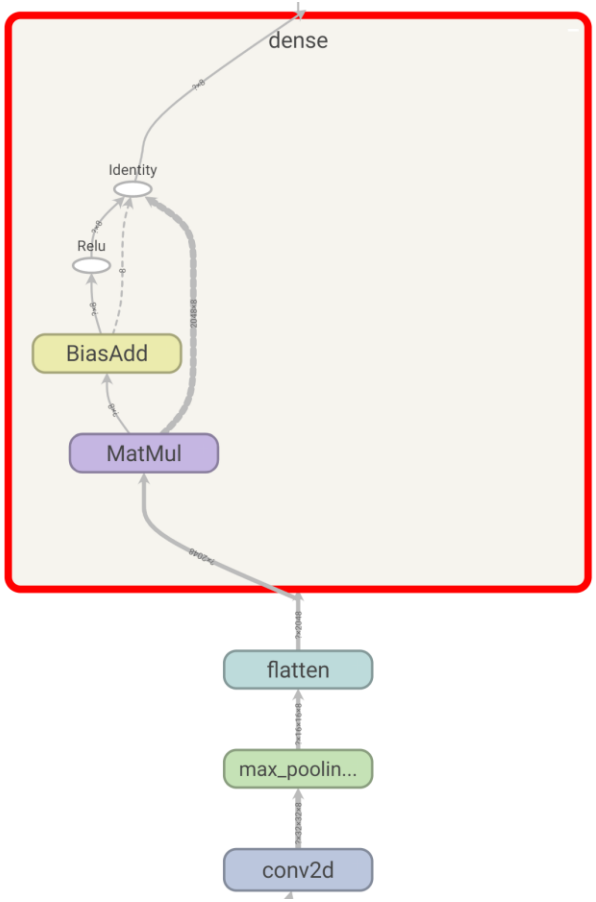
- Human accuracy on 100 airplane and automobile images: 93%

- Need autonomous recognition through camera image data quickly and at scale
- We assume machine learning, image recognition will be more accurate

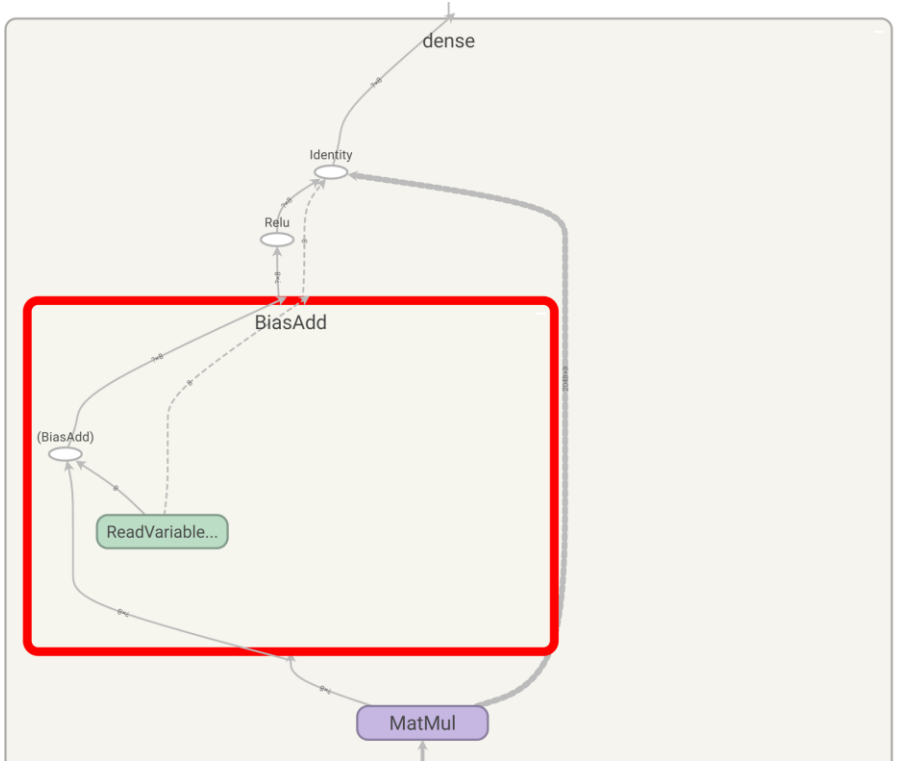
Base Model


metrics
loss
dense_1
dense_1_...
dense
flatten
max_poolin...
conv2d
conv2d_i...

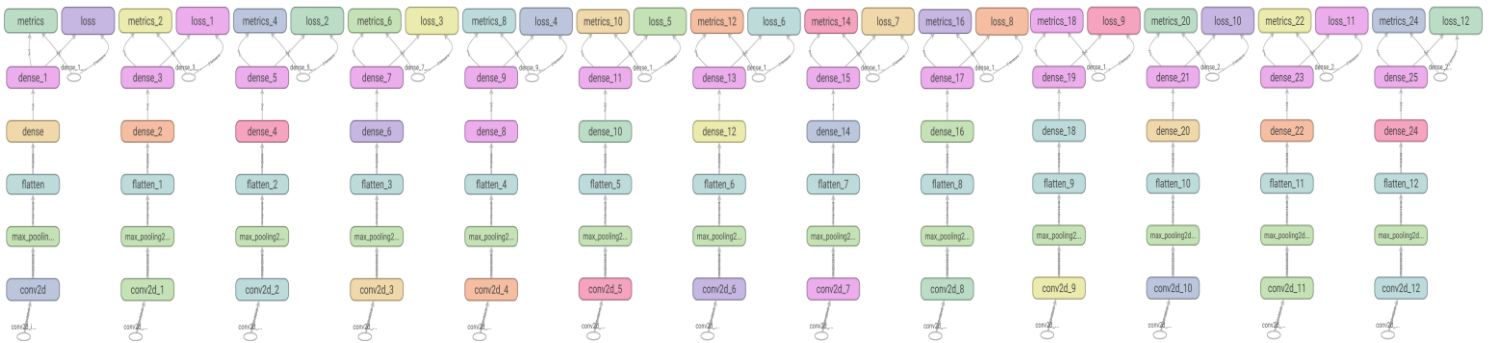
Base Model


dense
Identity
Relu
BiasAdd
MatMul
flatten
max_poolin...
conv2d

Base Model



Base Model, Expanded



Base Model

```
filters = 8
windowSize1 = (2,2)
windowSize2 = (2,2)
l2Decay = 0
leakyRelu = LeakyReLU(0)
denseUnits1 = 8
denseUnits2 = 1

model1 = Sequential()
model1.add(Conv2D(filters, windowSize1, padding='same', activation=leakyRelu,
                  kernel_regularizer=l2(l2Decay), input_shape=(32, 32, 3)))
model1.add(MaxPooling2D(pool_size=windowSize2))
model1.add(Flatten())
model1.add(Dense(denseUnits1, activation='relu'))
model1.add(Dense(denseUnits2, activation='sigmoid'))

model1.compile(optimizer='adam', loss='binary_crossentropy', metrics=['accuracy'])
```

Other Models Tested

```
filters = 8
windowSize1 = (2,2)
windowSize2 = (2,2)
l2Decay = 0
leakyRelu = LeakyReLU(0)
denseUnits1 = 8
denseUnits2 = 1
```
Val Accuracy = 88.5%

```
filters = 32
windowSize1 = (3,3)
windowSize2 = (3,3)
l2Decay = 0.0001
leakyRelu = LeakyReLU(0.0001)
denseUnits1 = 16
denseUnits2 = 1
```
Val Accuracy = 92.8%

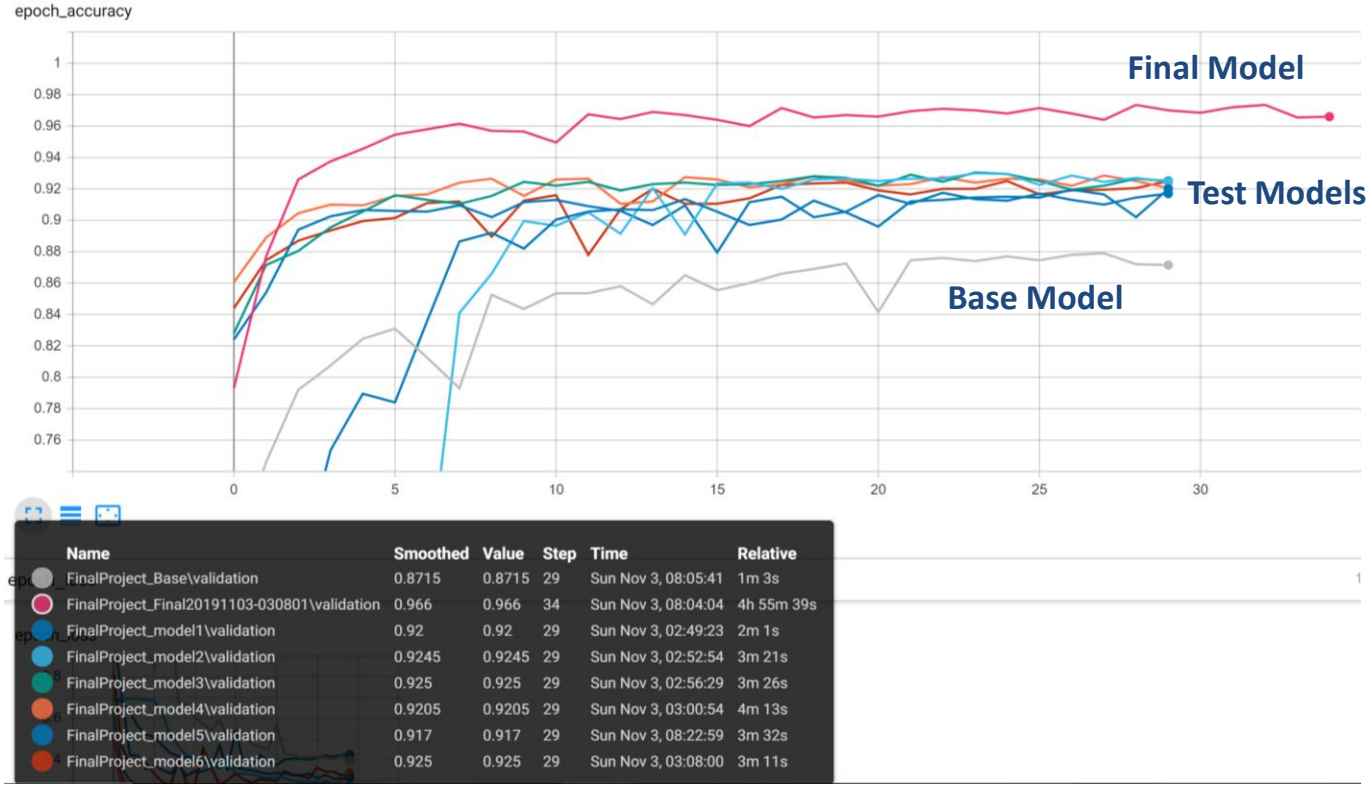
```
filters = 64
windowSize1 = (3,3)
windowSize2 = (3,3)
l2Decay = 0.0001
leakyRelu = LeakyReLU(0.0001)
denseUnits1 = 32
denseUnits2 = 1
```
Val Accuracy = 93.4%

```
filters = 64
windowSize1 = (3,3)
windowSize2 = (3,3)
l2Decay = 0.0001
leakyRelu = LeakyReLU(0.0001)
denseUnits1 = 64
denseUnits2 = 1
```
Val Accuracy = 93.3%

```
filters = 64
windowSize1 = (3,3)
windowSize2 = (3,3)
l2Decay = 0.001
leakyRelu = LeakyReLU(0.001)
denseUnits1 = 32
denseUnits2 = 1
```
Val Accuracy = 93.2%

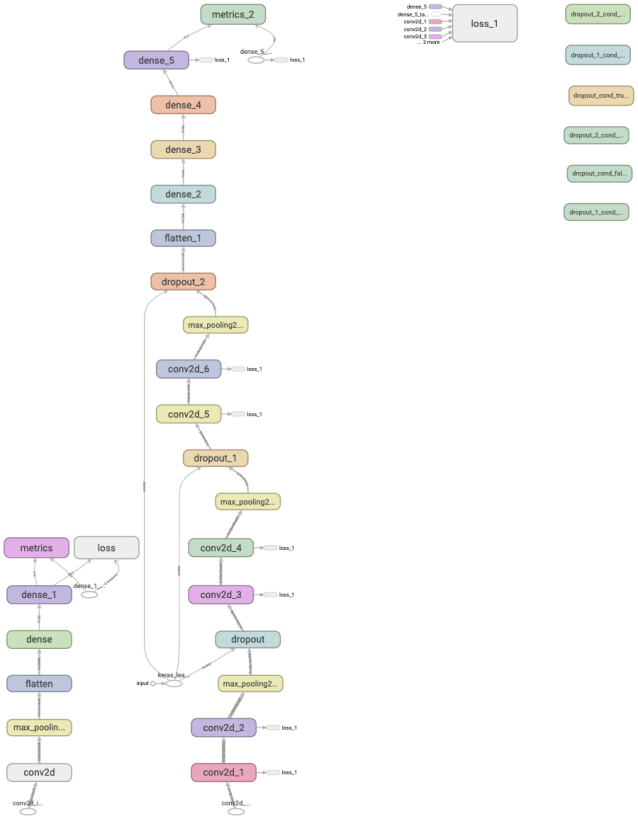
```
filters = 64
windowSize1 = (3,3)
windowSize2 = (3,3)
l2Decay = 0
leakyRelu = LeakyReLU(0)
denseUnits1 = 32
denseUnits2 = 1
```
Val Accuracy = 92.5%

Other Models Tested

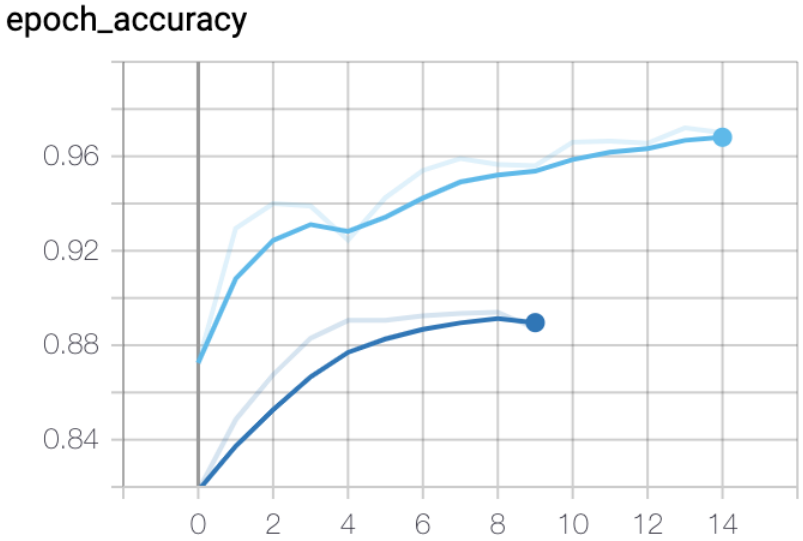


Name	Smoothed	Value	Step	Time	Relative
FinalProject_Base\validation	0.8715	0.8715	29	Sun Nov 3, 08:05:41	1m 3s
FinalProject_Final20191103-030801\validation	0.966	0.966	34	Sun Nov 3, 08:04:04	4h 55m 39s
FinalProject_model1\validation	0.92	0.92	29	Sun Nov 3, 02:49:23	2m 1s
FinalProject_model2\validation	0.9245	0.9245	29	Sun Nov 3, 02:52:54	3m 21s
FinalProject_model3\validation	0.925	0.925	29	Sun Nov 3, 02:56:29	3m 26s
FinalProject_model4\validation	0.9205	0.9205	29	Sun Nov 3, 03:00:54	4m 13s
FinalProject_model5\validation	0.917	0.917	29	Sun Nov 3, 08:22:59	3m 32s
FinalProject_model6\validation	0.925	0.925	29	Sun Nov 3, 03:08:00	3m 11s

Final Model Structure

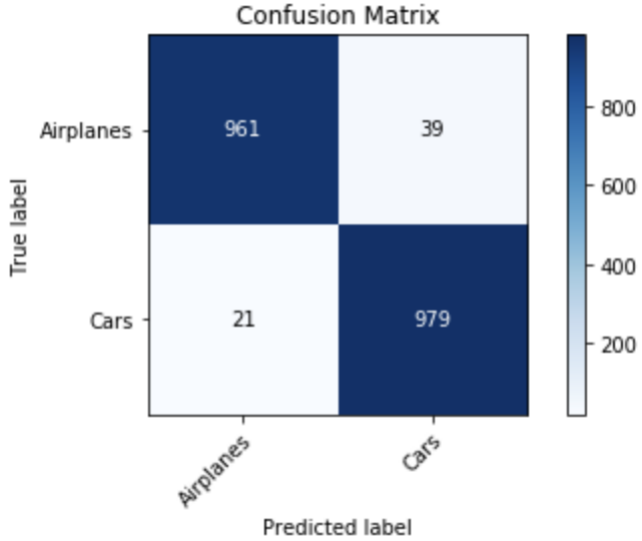


Final Model Performance



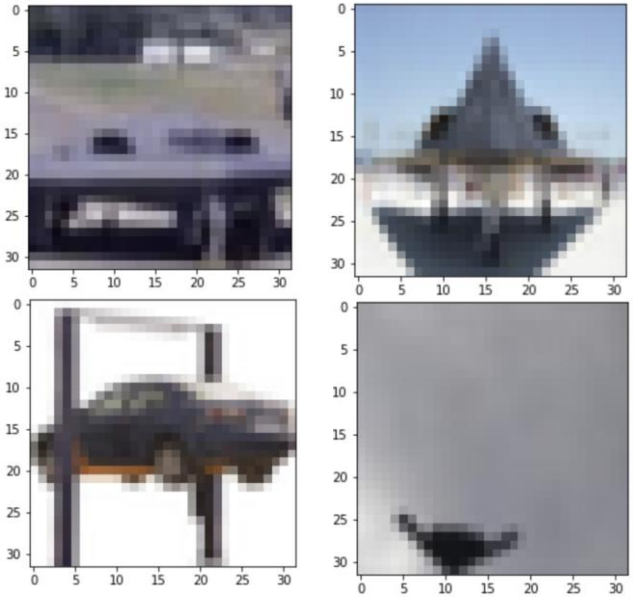
Name	Smoothed	Value
Base_model/validation	0.8896	0.887
FinalProject_Final/validation	0.9681	0.97

Final Model Details



- Model Structure
  - Convolutional layers (32, 64, 128 Neurons) window size 3x3
  - Followed by Pooling layers with 2x2 windows
  - 3 dropout layers of 0.2
  - 4 Dense Layers (64, 32, 16, 1)
  - 15 epochs, batch size 50
- Final Test Set Accuracy: 0.97

Misclassified Images



- 60 misclassified images out of 2,000
- Many are hard even for a human to determine or have features that are abnormal for the type of vehicle
- Moral of the story: stealth fighters can evade even the best CNN model

ATC Accident Prevention

- Cost effective solution utilizing existing camera infrastructure at airports
- Models yield high accuracy (97%) for the classification of objects on taxiways/runways
- Increased awareness and traffic management will reduce the number of incidents
- Misclassification of military aircraft is not an issue for commercial airports


VITA DUL-
CEDO SPES
UNIVERSITY OF
NOTRE DAME