Disijkstra

Alex Sissoko (ams2508), Pranav Shrestha (ps2958)

Open AI Caribbean Challenge: Mapping Disaster Risk from Aerial Imagery

Objective:

- Use satellite imagery to classify the roof material of identified buildings in St.Lucia, Guatemala, and Colombia





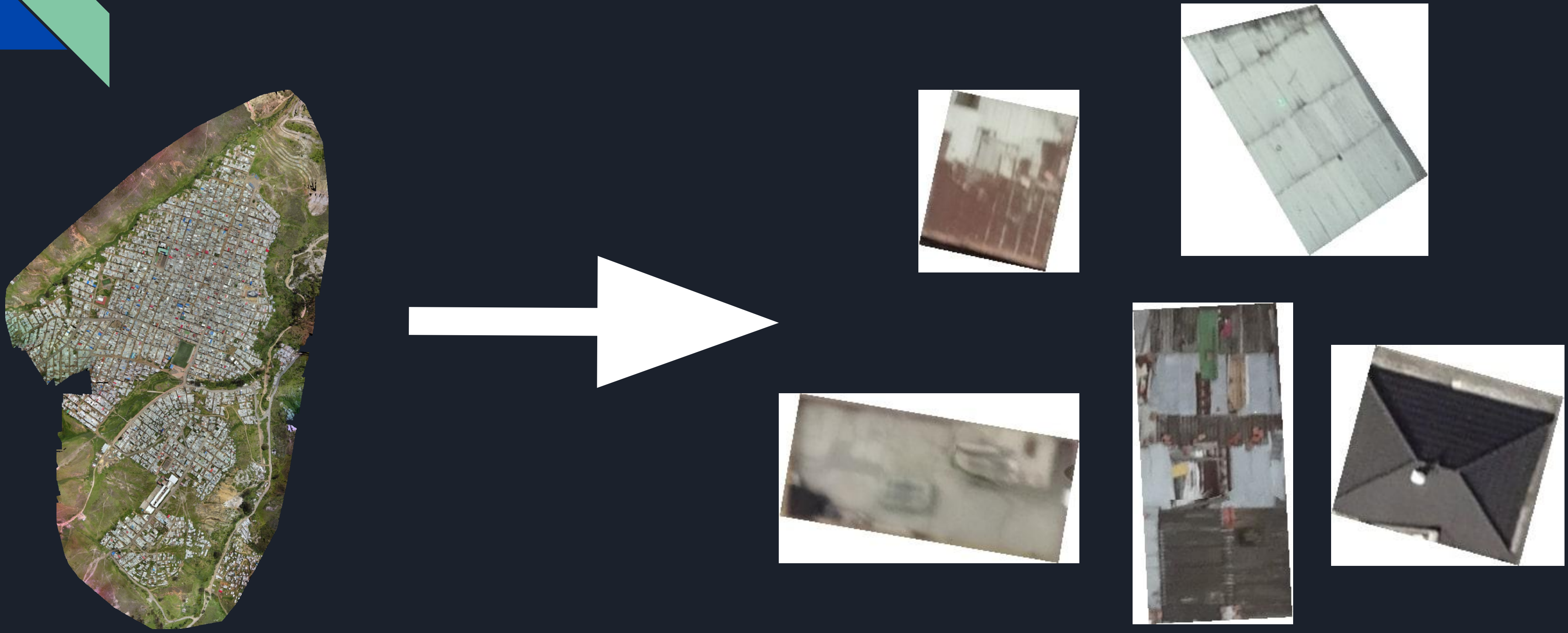
Dataset

- 7 high-res satellite images of different towns in 3 countries
- 5 classes of roof material
- Labels also classified as verified (14870) or unverified (7683)

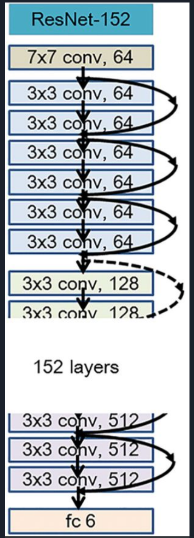
Country Name	Area Name	Image resolution
Colombia	borde_rural	(52318, 31315)
Colombia	borda_soacha	(40159, 45650)
Guatemala	mixco_1_and_ebenezer	(27054, 26641)
Guatemala	mixco_3	(26066, 19271)
St Lucia	castries	(50027, 62570)
St Lucia	dennery	(21184, 41534)
St Lucia	gros_islet	(53492, 90729)

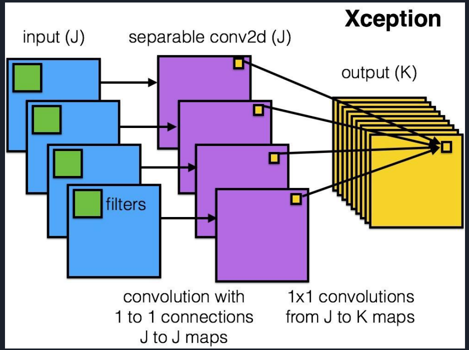
Roof material	Description	Count
concrete_cement	Roofs are made of concrete or cement	1518
healthy_metal	Includes corrugated metal, galvanized sheeting, and other metal materials	14817
incomplete	Under construction, extremely haphazard, or damaged	669
irregular_metal	Includes metal roofing with rusting, patching, or some damage. These roofs carry a higher risk	5241
other	Includes shingles, tiles, red painted, or other material	308

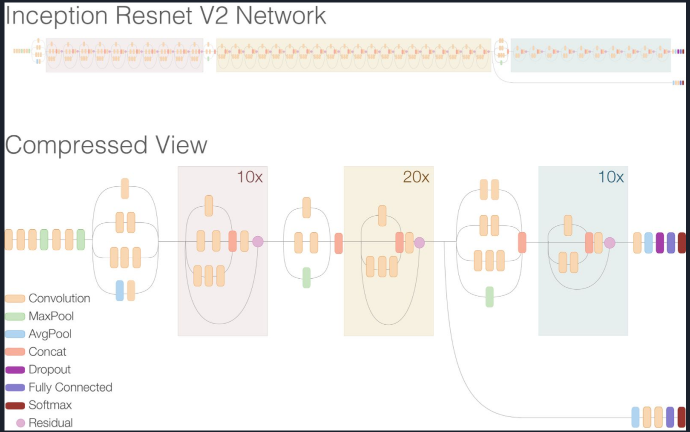
GeoJSON Data Parsing

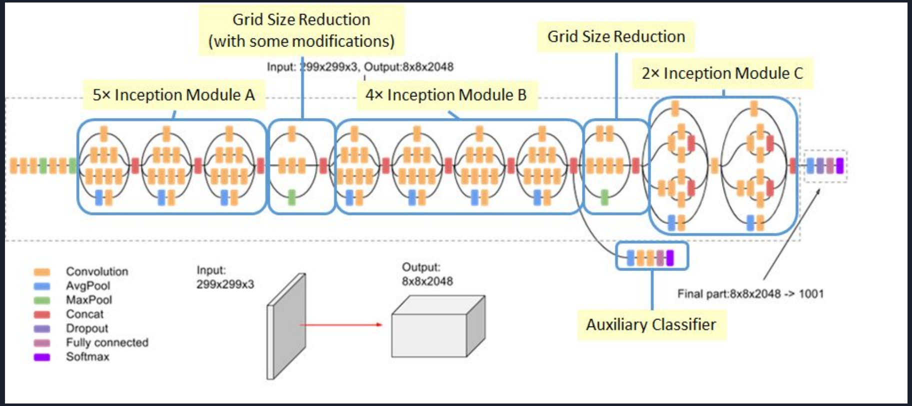


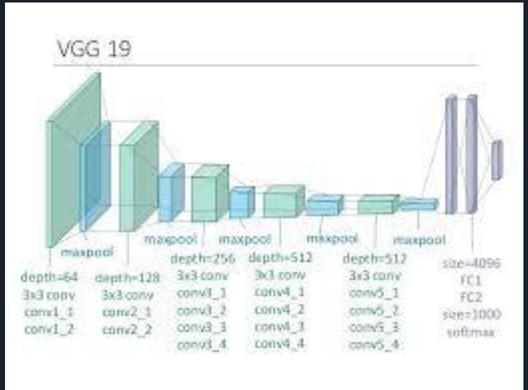
Benchmarking ImageNet models











Benchmarking in Numbers

Model	Training loss	Training accuracy	Validation loss	Validation Accuracy	Competition Benchmark
VGG19 (no_pretrain)	0.7435	0.7328	0.6712	0.7540	1.9042
VGG19 (pretrained)	0.6727	0.7600	0.6226	0.7736	2.1110
Resnet (no_pretrain)	1.1346	0.4989	1.1022	0.5030	1.9691
Resnet (pretrained)	0.9969	0.5981	0.9387	0.6161	1.7969
ResnetV2 (no_pretrain)	0.6463	0.7580	0.6256	0.7641	1.6584
ResnetV2 (pretrain)	0.6129	0.7651	0.5889	0.7799	2.6665
Inception (no_pretrain)	0.8994	0.6193	0.7876	0.6786	2.0023
Inception (pretrain)	0.8438	0.6668	0.7302	0.7305	1.7199
Xception (not_pretrain)	0.8193	0.6829	0.8091	0.6915	2.2347
Xception (pretrain)	0.8159	0.7003	0.7093	0.7433	1.8669

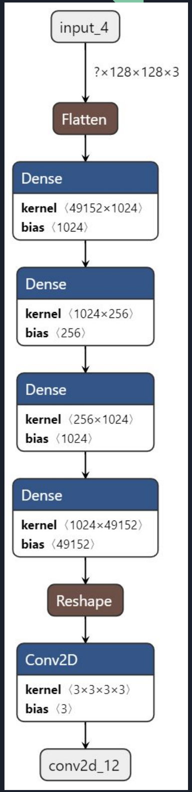
A Simple CNN

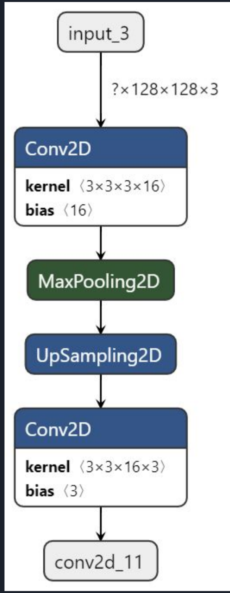
```
model = Sequential([
    Conv2D(32, (5, 5), input_shape=(img_width, img_height, 4)),
    Flatten(),
    Dense(256, activation='relu'),
    Dropout(0.5),
    Dense(5, activation='softmax')
])
```

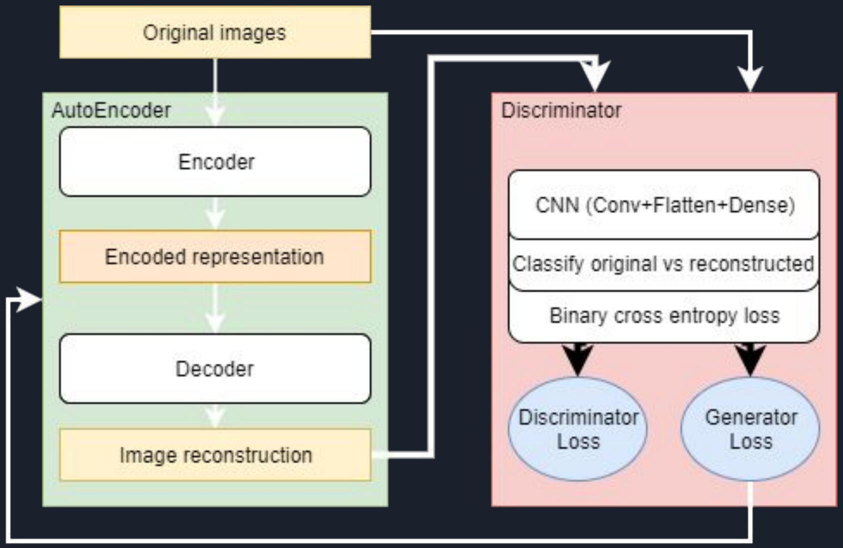
Model	Training loss	Training accuracy	Validation loss	Validation Accuracy	Competition Benchmark
SimplCNN	1.0480	0.5749	1.0125	0.5794	1.1629

AutoEncoders

- Main idea was for dimensionality reduction for
    - Apply principal component analysis (PCA) to visualize differences in classes
    - Apply regression for classification.
- Implementations include MLP autoencoders, Shallow/Deep CNN autoencoders and GANs.

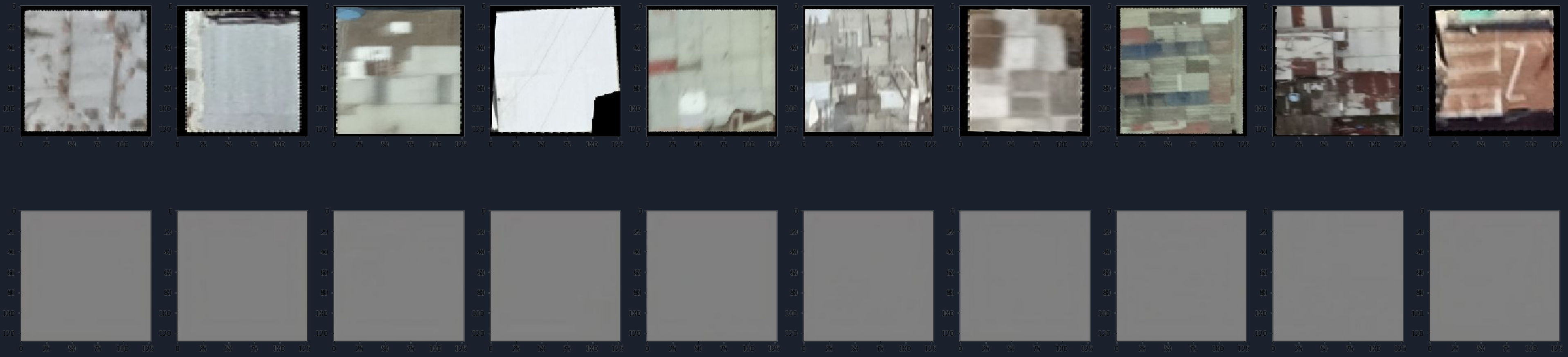


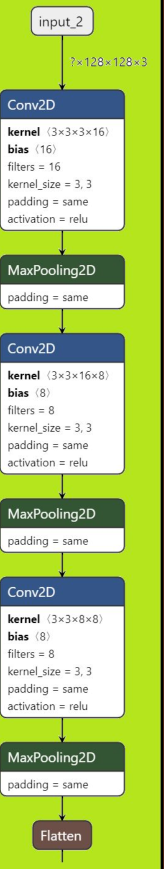


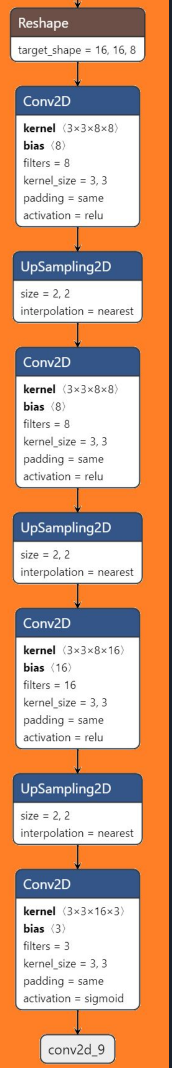


AutoEncoders - Two errors

- Using ‘adadelta’ (or ‘adagrad’) as optimizers
    - Always stuck as local optima, which was simply to guess the averaged pixel values
- Non-aligned images
    - High variations in the images making it impossible for the autoencoder to learn







Improved Autoencoder

- Deep CNN architecture (Conv2D-MaxPool2D -> Conv2D-UpSampling2D)
- Aligned images to reduce possible image space
- Optimizer used : Adam/Nadam
- MSE Loss achieved : 0.003
- Outcome:
  - Unfortunately, regression did not work. The details lost due to the auto-encoder made it impossible to distinguish corrugated metal panels from plain concrete.
  - However, it was one of the factors that led to our alignment approach

Unbalanced Data

- Challenges: Model predictions are even further unbalanced.
- Approaches: Oversampling, Undersampling, Class weights

Training Data Distribution

Class	concrete cement	healthy metal	incomplete	unhealthy metal	other
Training Instances	1110	5905	535	4193	155

Training Data Predictions Distribution

Class	concrete cement	healthy metal	incomplete	unhealthy metal	other
Training Instances	512	6021	23	5279	63

Metal vs nonmetal prediction

- Split dataset into metals and non-metals
    - Idea was to separate metals first, then build secondary classifier for remaining classes.
- Did not train at all. Always predicted metal since distribution was imbalanced
    - Metal had > 84% of dataset
- Attempted fixes:
    - Imbalanced fixes: Undersampling, class weights
    - Edge detection (Sobel filters, image gradients)

Mean Distribution Prediction

- Simply predicted the mean class distribution for all test images
    - Got a 1.24 on public leaderboard
- Predicted mean class distribution per town for all test images
    - Got a shocking 1.01 on public leaderboard

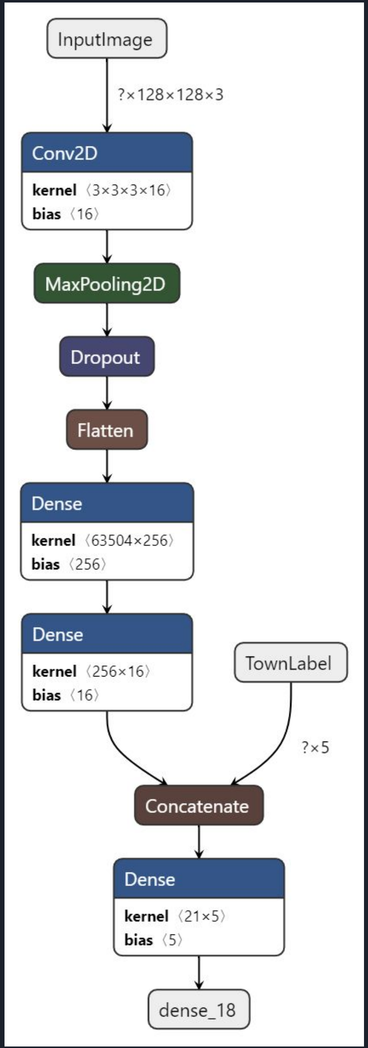
```
# Read in all training labels. It exists in a one-hot encoded format
# Simply taking the mean gives us class distributions for the whole set
train_labels = pd.read_csv("train_labels.csv", index_col='id')
train_labels.iloc[:, 1:].to_numpy().mean(0)
# array([ 0.067,  0.657,  0.030,  0.232,  0.014])

# Create a submission file with every image classified as the mean
submission   = pd.read_csv('submission_format.csv', index_col='id')
submission.iloc[:, :] = train_labels.iloc[:, 1:].to_numpy().mean(0)
submission.head()

submission.to_csv("result.csv")
```

Image Alignment

- Basic idea is that corrugated metal is the most important feature
- Most buildings apply corrugated metal roofs parallel to one border.
- Instead of having the model check every orientation of corrugated edges, align the buildings so that it only has to check for corrugation horizontally or vertically.
- Used cv2.minAreaRect(coords)

Dual Input Model

- We know town distributions are important.
- Basic idea: Include town labels as input into the model.
- Initially included many improvements from before
  - Converting to HSV for brightness invariance
  - Image gradients are additional image channels
- Overfit on the training data! 1.0 accuracy achieved
  - Added Dropout and decreased model complexity



Final Results

Best Model as of 6pm

- Image gradient channels
- HSV color space conversion
- 3 Conv2D+MaxPool+Dropouts
- Flatten + Dense
- Augmented with town label

Woohoo! We processed your submission!

Your score for this submission is:

0.6948

Woo! I scored 0.6948 on 'Open AI Caribbean Challenge: Mapping Disaster Risk from Aerial Imagery'!

Submissions

BEST	CURRENT RANK	# COMPETITORS	SUBS. MADE
0.6948	60	1338	3 of 3