First-Floor Elevation Prediction

Copello | Fliege | Guardi

Project Background / Problem Statement

- The only way for insurance providers to obtain the FFH is by examining an elevation certificate for a given property.
- FFH is an important factor in determining the likelihood and severity of damage being caused by a flood event. We define flood as rising surface waters or an overflood of inland or tidal waters. Ground elevation is determined in feet above sea level.
- The model needs to predict first-floor heigh (FFH), which is the number in feet of the first living floor above ground elevation, for risk modeling purposes. This variable is an upstream input of several risk models.

WNC INSURANCE SERVICES

Business Value

Cost of Physical Measurement
Estimated $150-$200 for elevation certificate using physical measurement

State of Current Automation Efforts
- True Flood provides a solution that has performed poorly ($5 per prediction per feature predicted)
- Additional companies working on solutions but are in development and unavailable to our client

Impact for WNC Insurance
- WNC has a portfolio of roughly 100,000 residential properties in flood prone areas
- If the model predicts 10% of an average sized portfolio this represents an estimated $1.5 Million in savings

Process High Level

Independent Variables

Google StreetView Image Data
- 640 x 640 Color Image
- Elevation Metadata
- Camera Angle Metadata
- Geolocation Data for Vehicle and Property

YOLOv5

USGS Topographical Data
- Base Elevation
- Geolocation Data

Florida Division of Emergency Management
- Elevation Data
- First-Floor Height

Feature Engineering

Dependent Variables

Image Bounding Box Data
- Identified House / Probability
- Identified Front Door / Probability
- Identified Garage Door / Probability
- Reference Coordinates

Engineered Features

Additional Model Input
- Measured Physical Features

Assumed standard height of residential entry

Calculated First Floor Height

Data - Provided

Initial Data From Client
Data Quality Questionable

1,300 Properties Represented

- Physical Address
- Latitude/Longitude
- Lowest Adjacent Grade
- Highest Adjacent Grade
- First Floor Height (Data Integrity Issues)
- Multiple States
- Flood Zone Information
- Base Flood Elevation
- Other Geographic Features

Revised Data From Client
Optical Character Recognition

3,814 Total Properties Represented

- Physical Address
- Latitude/Longitude
- Building Diagram Number
- Lowest Adjacent Grade
- Highest Adjacent Grade
- First Floor Height

Data - Derived

Google Data

- Latitude and Longitude
- Fuzzy Address Search
- Location of street view car for image orientation
- Image verification
- 640 x 640 png
- Dynamically computed heading

USGS Data

Florida Division of Emergency Management

Research

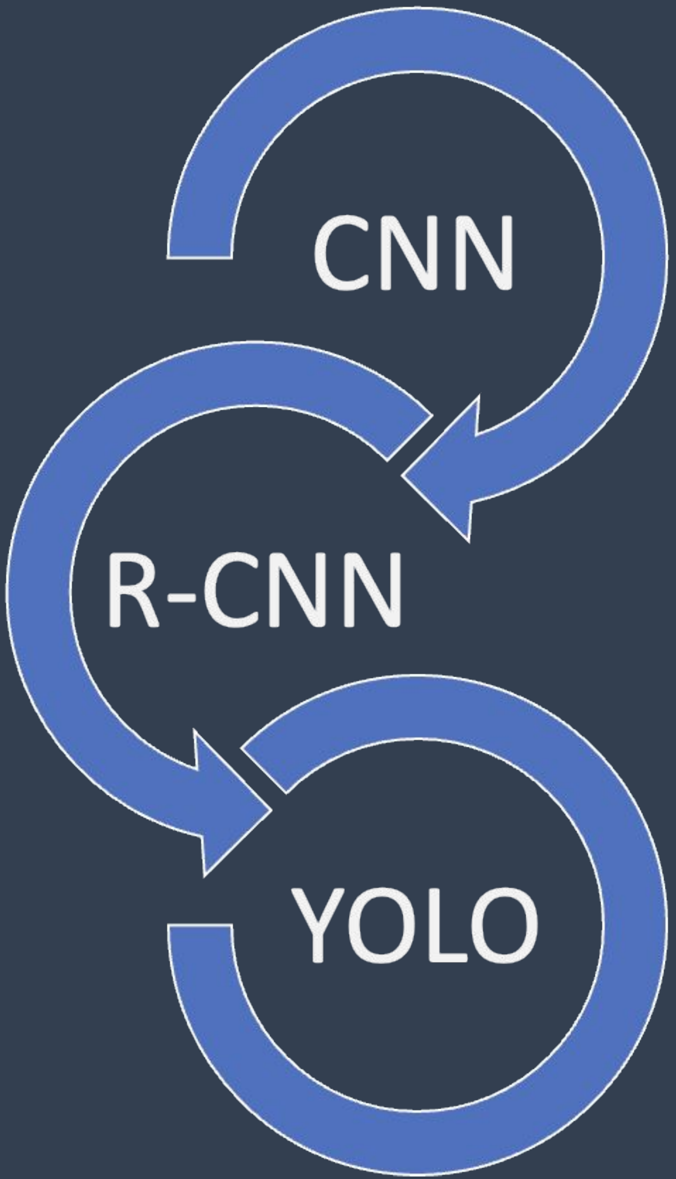
Google Street View & Computer Vision
Access to ground level images from around the world at a low price

FFH Predictions
Galveston, TX and Hampton Roads, VA public projects to predict FFH

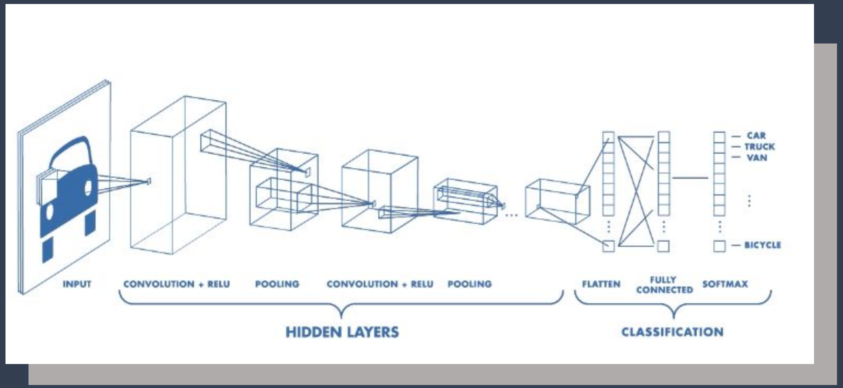
Our Work

Insurance
Claims and damage measurement

Transfer Learning Evolution



- Sliding window, pixel by pixel process
- Suitable for single object detection or image classification
- Slow training, poor performance on small exploratory dataset

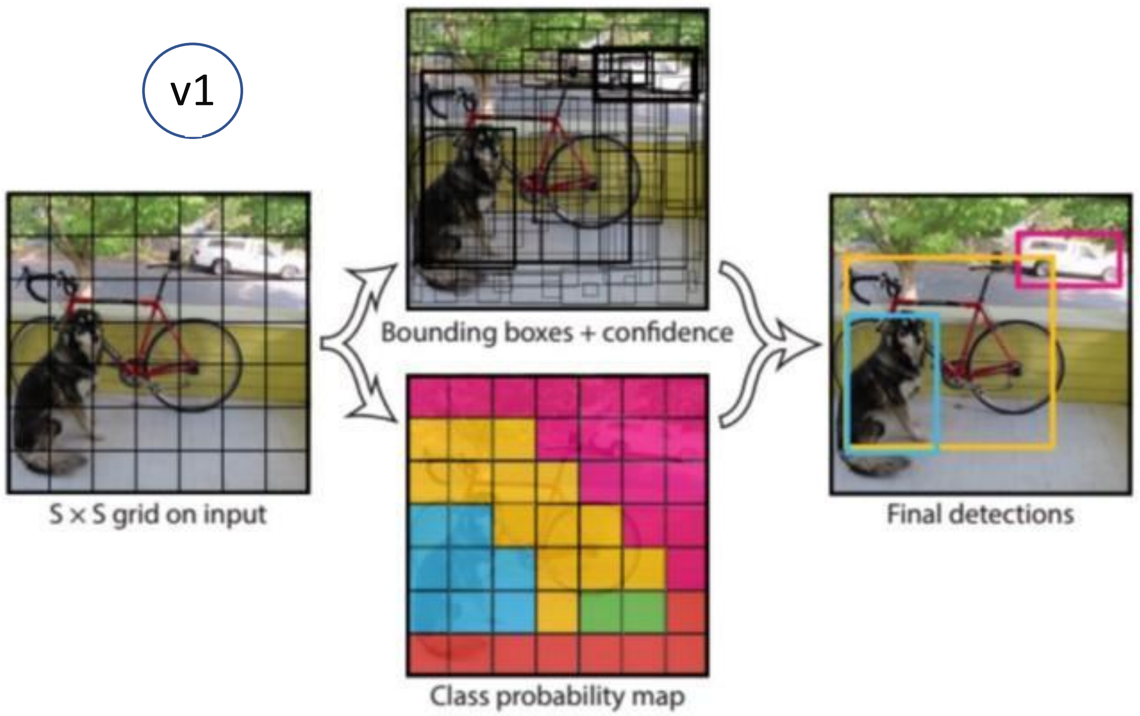


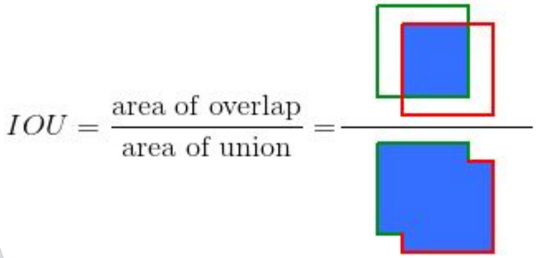
- Region proposal allows for multiple object detection and instance segmentation (Mask R-CNN)
- Captures House, Front Door and Garage in two step process

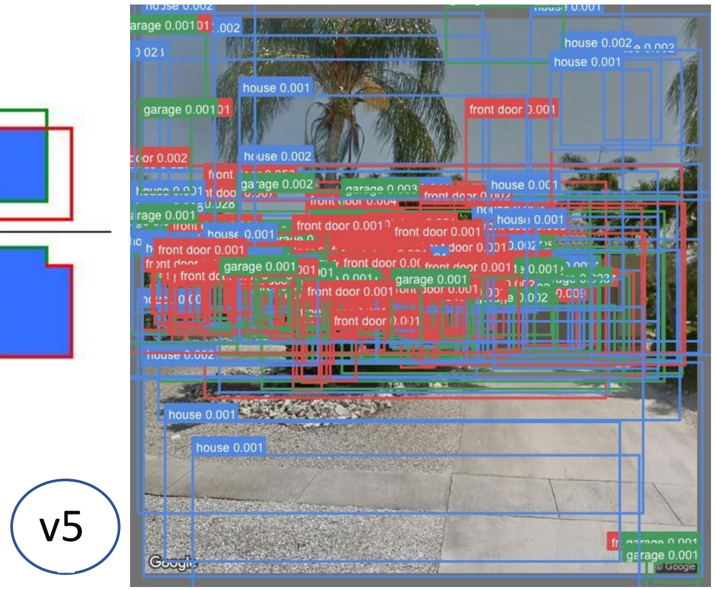
- Takes in entire image at once and implicitly encodes context
- Combines object classification and bounding box regression in one step
- Fast training, fast inference, generalizing ability

You Only Look Once (YOLO v5)

- Developed by Joseph Redmon for v1-3
- YOLOv5 developed by Glenn Jocher of Ultralytics, contributors to YOLOv3 pytorch implementation (2,379 commits!)
- Improvements from v1 to v5 include better object detection for tightly grouped objects when more than 1 object is in a cell
- YOLOv5 uses Mosaic Augmentation (developed by Jocher) and integrates with python Albumentations package





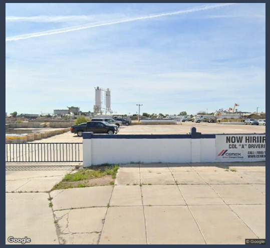


Label and Train

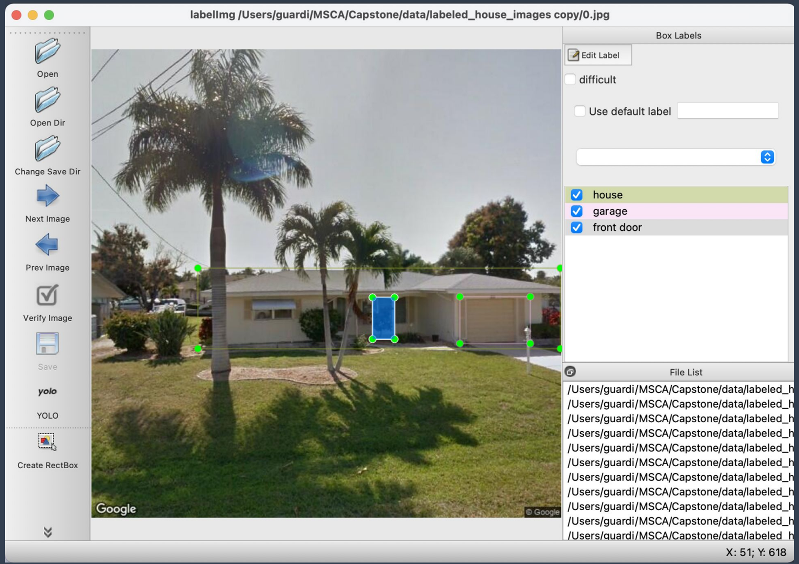
Google API call for Street-View Image

Download image

Label House/No House



Enhanced Labeling



Class	x_center	y_center	Width	Height
0	0.504687	0.490625	0.921875	0.306250
2	0.635938	0.537500	0.046875	0.118750
1	0.223438	0.554688	0.193750	0.146875

YOLOv5

Model Training

- Fine-tune YOLO on labeled images and custom categories
- Training dataset 948 images, validation 122
- Best model weights at 500 epochs
- Training time 3-5 hours/run on Google Colab

Metrics and Results

Validation Results

Class	Images	Labels	P	R	mAP@.5
all	122	270	0.541	0.525	0.499
house	122	155	0.874	0.896	0.922
front door	122	65	0.313	0.2	0.232
garage	122	50	0.436	0.48	0.342

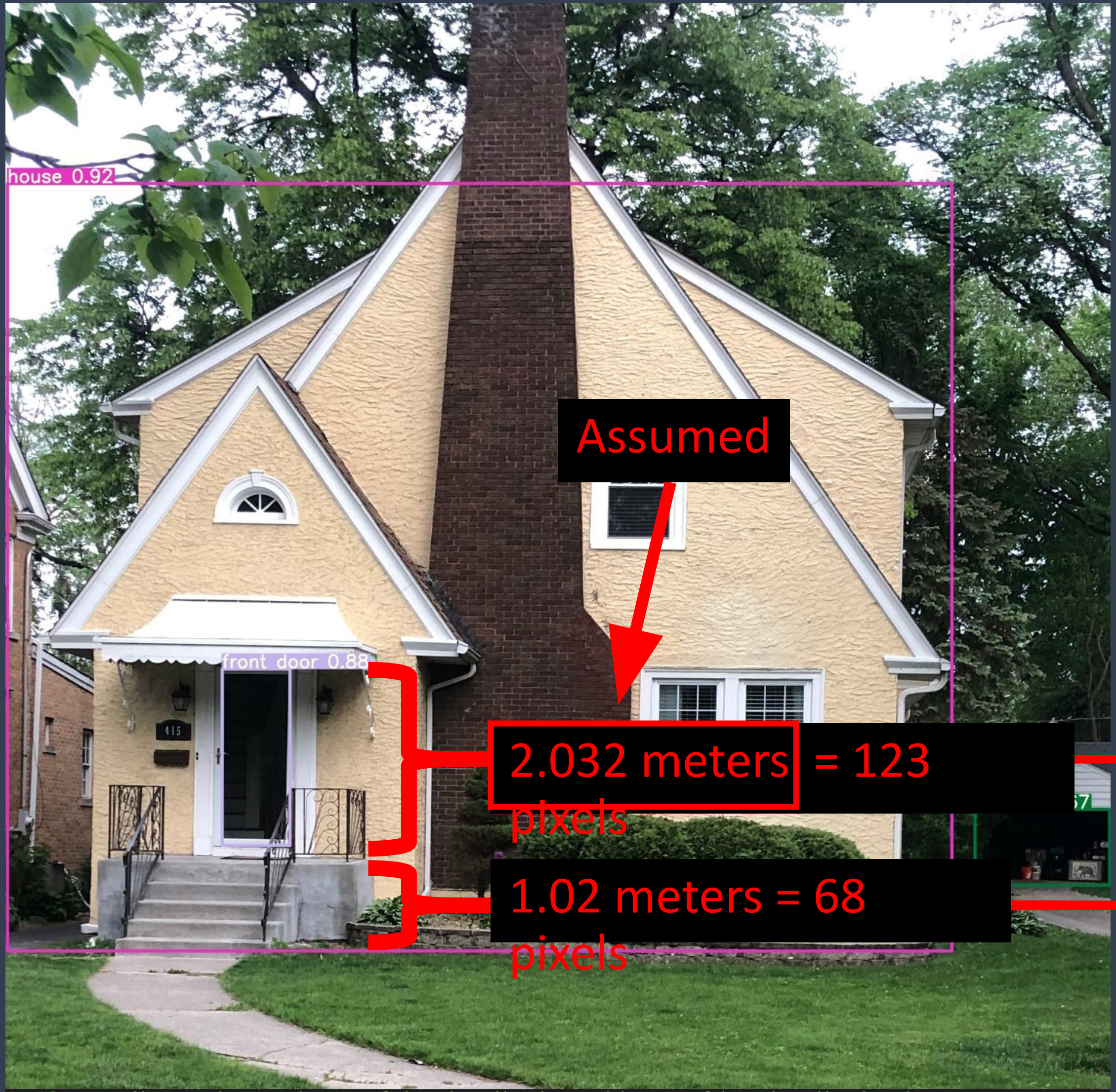
- The more labeled data the better the results
- Label imbalance in favor of houses
- Larger models showed better results (training time 2.5x longer)

Tradeoffs: Speed vs Accuracy

- Iteration speed increases the opportunity to evolve the product with new geographic areas, house styles
- Edge case investigation > Optimizing for higher accuracy
- “Training humans to visually inspect a bounding box with IOU of 0.3 and distinguish it from one with IOU 0.5 is surprisingly difficult.” - Olga Russakovsky

Weights & Biases

FFH Measurement




```
#Compute distance between bottom of door bounding box and bottom of home bounding box in meters
def ffh_from_box(h_y_center, h_height, d_y_center, d_height, standard_door_height):
    dis = (h_y_center + (h_height/2) - (d_y_center + (d_height)/2)) * standard_door_height / d_height
    return (dis)
```

- YOLO Inference result is an image with bounding boxes and a text file with the label and location
- Assume standard door height = 2.032 meters
- By assuming a standard door height in meters, and measuring it in pixels, this becomes a simple unit conversion problem
- If the difference between the bottom edge of the home and the bottom edge of the door is x pixels, how many meters is it equal to?
- Depth matters greatly

Findings

Model	Sample Size	% of Predictions (Total Predicted)	RMSE (m)	MAE (m)
YOLOv5 + FFH Calculation Methodology	617	26% (161)	0.726	0.0309
Tree Ensemble via H2O + Elevation Certificates	617	100% (617)	1.42	0.747
Baseline Heuristic Estimate	617	100% (617)	2.875	1.922

Given that elevation certificate data is available, documented property features are excellent predictors. Unfortunately, elevation certificates are expensive (and already include FFH)

Baseline model main predictor is building diagram type, unknowable at inference time

Conclusions

Our application unlocks previously unattainable value from an enormous store of unstructured data at the cost of API calls and minimal compute

Data Limitations

- Images are hard to source
- Unethical to scrape

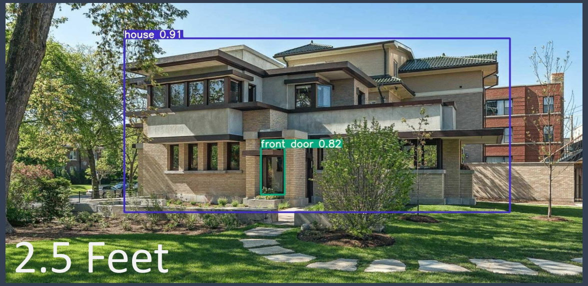
- Portfolio Limitations
- Portfolio homogeneous
- Model generalizes well outside of Florida

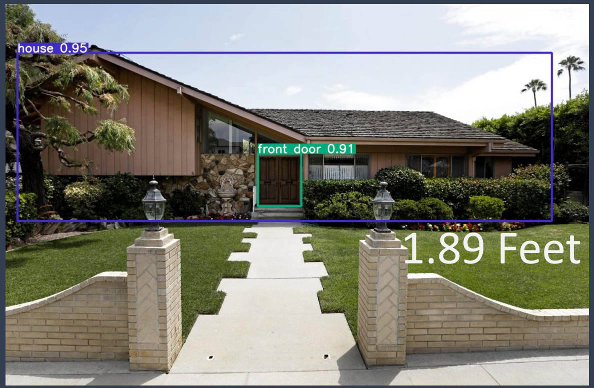
Estimated Savings

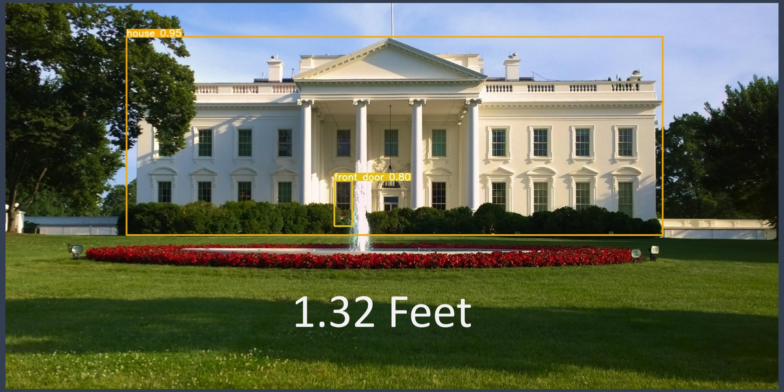
- Potential value with better data = total properties x elevation certificate cost
- Estimated value of data unlocked for WNC between 1 and 1.25 million dollars.

Risk Model Impact (if available)

- WNC to Provide Risk Impact







Recommendations

Acquire Better Data

- Higher quality data
- Alternate Data Sources
    - Zillow
    - Redfin
    - Realtor.com
    - Images taken by user

Augment Data

- GIF of property with average
- Monocular or binocular depth estimation

Move Software to Production

- Streamlit app
- Batch Estimation App

Appendix

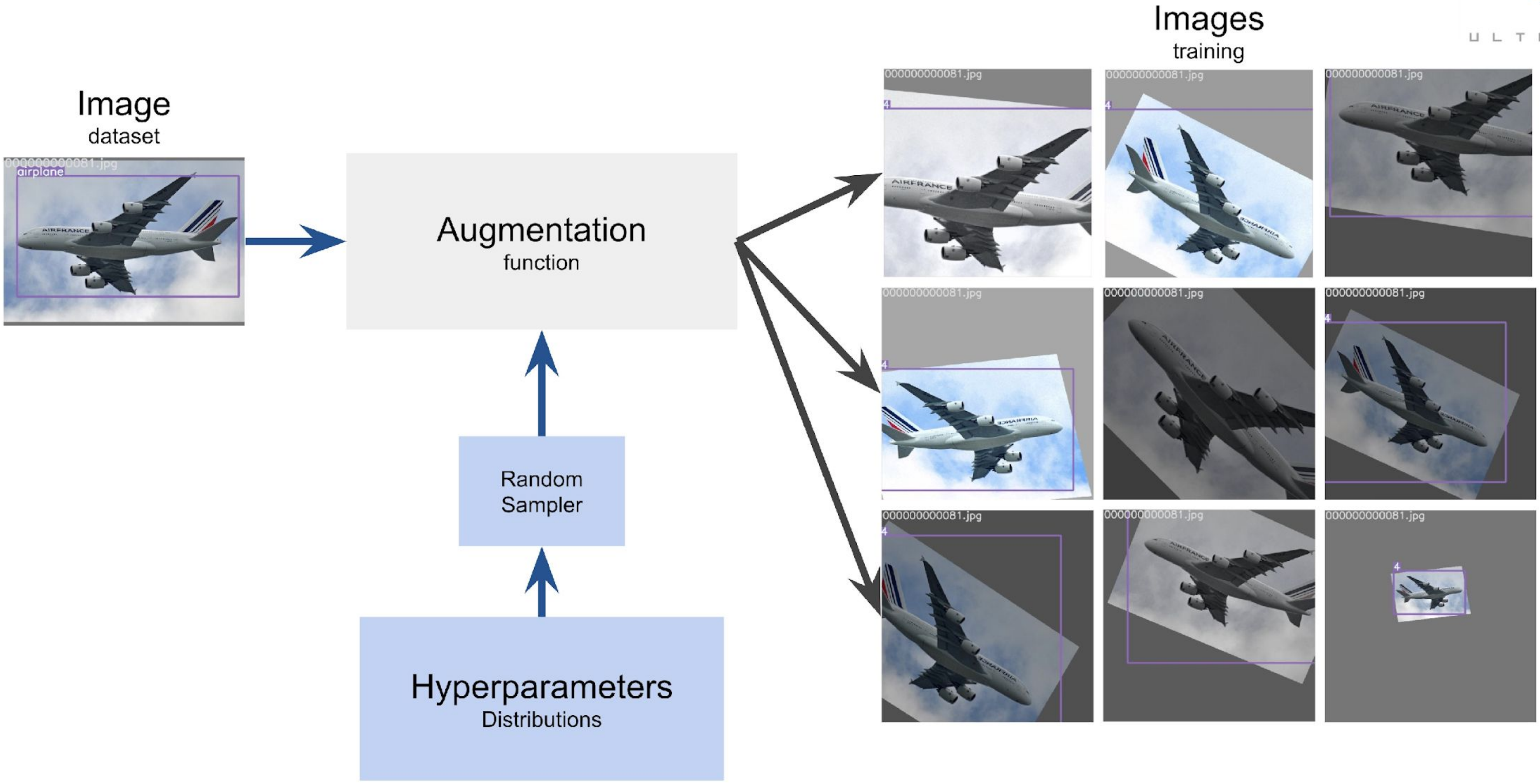
Click to add text

YOLOv5 Augmentations

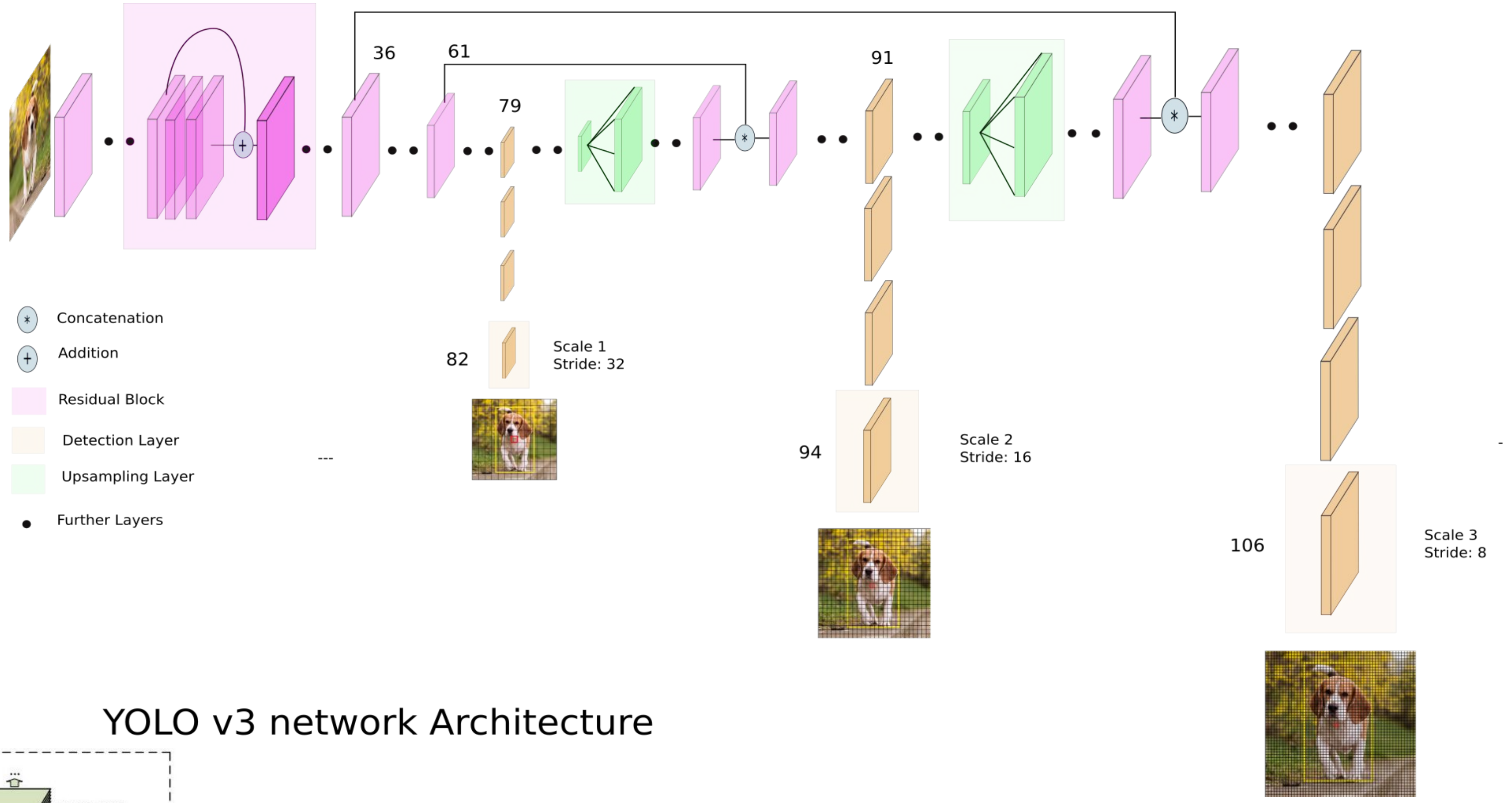
- High degree of scaling augmentation
- Mosaic data loader
- Color space augmentation
- V3 would upscale the entire image (memory intensive)
- V5 image same size (640)

```
hsv_h: 0.015  # image HSV-Hue augmentation (fraction)
hsv_s: 0.7  # image HSV-Saturation augmentation (fraction)
hsv_v: 0.4  # image HSV-Value augmentation (fraction)
degrees: 0.0  # image rotation (+/- deg)
translate: 0.1  # image translation (+/- fraction)
scale: 0.5  # image scale (+/- gain)
shear: 0.0  # image shear (+/- deg)
perspective: 0.0  # image perspective (+/- fraction), range 0-0.001
flipud: 0.0  # image flip up-down (probability)
fliplr: 0.5  # image flip left-right (probability)
mosaic: 1.0  # image mosaic (probability)
mixup: 0.0  # image mixup (probability)
```

YOLOv5 Image Augmentation

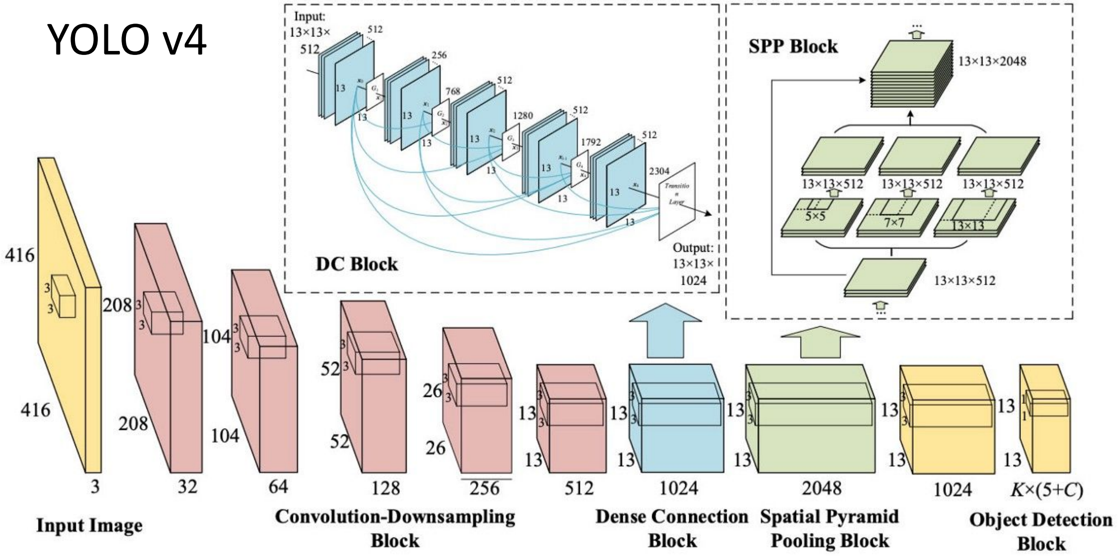


Obligatory Architecture Slides

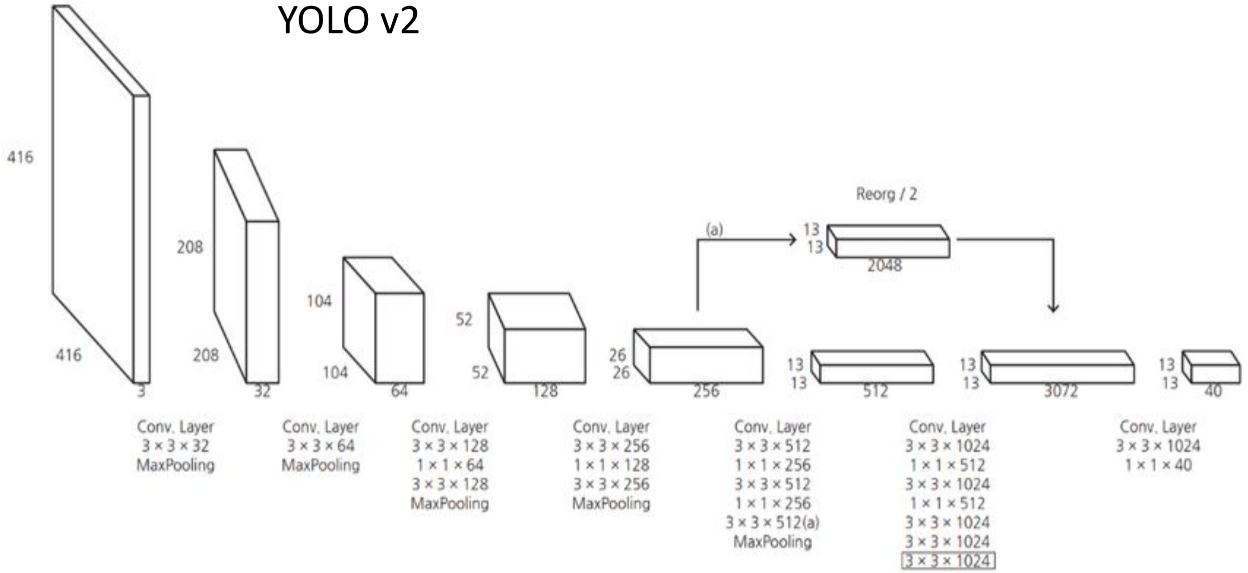


YOLO v3 network Architecture

YOLO v4



YOLO v2



•https://towardsdatascience.com/yolo-v3-object-detection-53fb7d3bfe6b

YOLO Hyperparameters

- lr0: 0.01 # initial learning rate (SGD=1E-2, Adam=1E-3)
- lrf: 0.2 # final OneCycleLR learning rate (lr0 * lrf)
- momentum: 0.937 # SGD momentum/Adam beta1
- weight_decay: 0.0005 # optimizer weight decay 5e-4
- warmup_epochs: 3.0 # warmup epochs (fractions ok)
- warmup_momentum: 0.8 # warmup initial momentum
- warmup_bias_lr: 0.1 # warmup initial bias lr
- box: 0.05 # box loss gain
- cls: 0.5 # cls loss gain
- cls_pw: 1.0 # cls BCELoss positive_weight
- obj: 1.0 # obj loss gain (scale with pixels)
- obj_pw: 1.0 # obj BCELoss positive_weight
- iou_t: 0.20 # IoU training threshold
- anchor_t: 4.0 # anchor-multiple threshold
- # anchors: 0 # anchors per output grid (0 to ignore)
- fl_gamma: 0.0 # focal loss gamma (efficientDet default gamma=1.5)