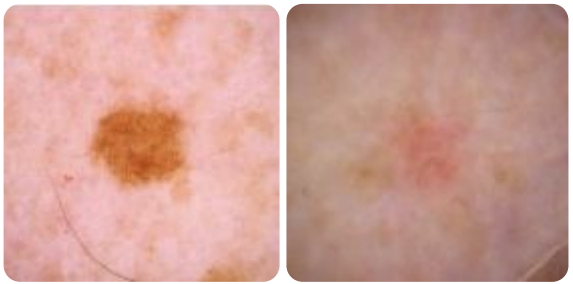
Classifying Melanoma

What Is the Melanoma Dataset?

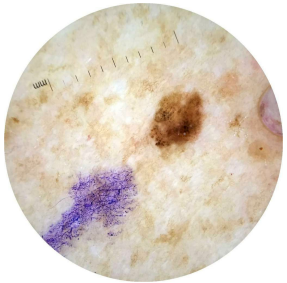
- Annual competition on Kaggle
  - Lots of highly skilled veterans!
- Expert Judgement: 74.1% sensitivity and 60% specificity by dermatologists
- Challenges
  - Large Data Set: 35 GB+ of JPEGs
  - Unbalanced classes: Only 500 positive examples out of n=30,000+)
  - Heterogeneous format: Inconsistent image quality/shape/brightness/orientation
  - Visually ambiguous classes

Can you tell which one is Melanoma?



Sexy hair...

Round crop!?



Baseline Models

Baseline Models

- Logistic Regression
- Support Vector Machine

Baseline Results

Logistic Regression

	precision	recall	f1-score	support
0	0.98	1.00	0.99	6441
1	0.00	0.00	0.00	117
accuracy			0.98	6558
macro avg	0.49	0.50	0.50	6558
weighted avg	0.96	0.98	0.97	6558

SVM (Linear Kernel)

	precision	recall	f1-score	support
0	0.98	0.99	0.99	6440
1	0.07	0.05	0.06	116
accuracy			0.97	6556
macro avg	0.53	0.52	0.52	6556
weighted avg	0.97	0.97	0.97	6556

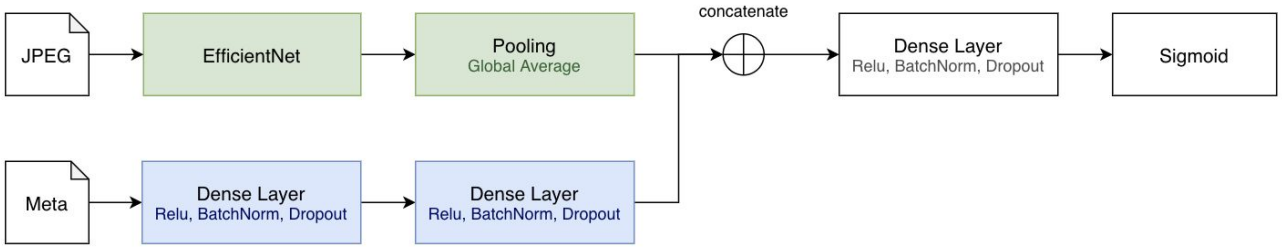
Notes:

- 10 Epochs
- Sigmoid Threshold = 0.5
- No Regularization

Neural Networks

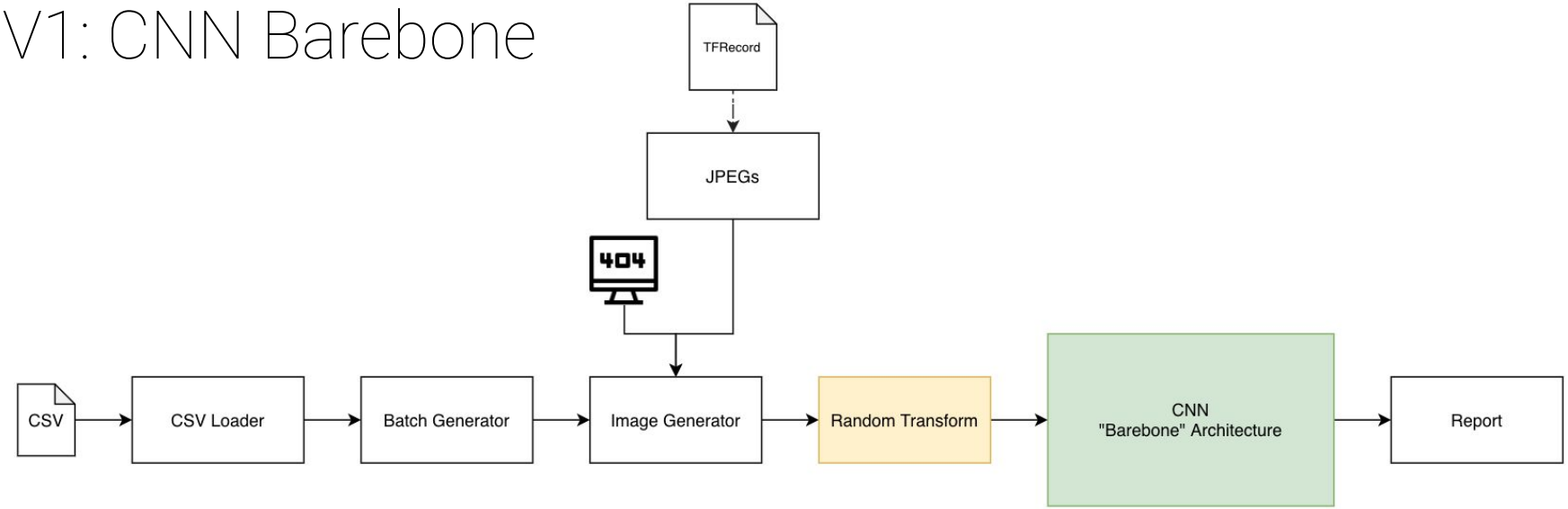
Three Attempts

- V1: CNN Barebone
- V2: CNN Advanced
- V3: CNN (Image) + Neural Network (Metadata)



- Bonus EDA: Can neural network learn?

V1: CNN Barebone



Architecture 1

- Input
- Conv. 3x3x64, ReLu
- Conv. 3x3x32, ReLu
- Flatten
- Dense, Softmax

Architecture 2

- Input
- Conv. 3x3x75, ReLu
  - Pool 2x2
  - Dropout 0.15
- Conv. 3x3x125, ReLu
  - Pool 2x2
  - Dropout 0.15
- Flatten
- Dense, Softmax

V1: Miserable Failure

First run...

	precision	recall	f1-score
0	0.50	1.00	0.67
1	0.00	0.00	0.00
accuracy			0.50
macro avg	0.25	0.50	0.33
weighted avg	0.25	0.50	0.33

After countless hours of tweaking...

	precision	recall	f1-score
0	0.53	0.96	0.68
1	0.78	0.14	0.24
accuracy			0.55
macro avg	0.65	0.55	0.46
weighted avg	0.65	0.55	0.46

And... a full epoch takes ~1 hour to complete...
Scrapped 1,000+ lines of codes... Ouch!

V2: CNN Advanced

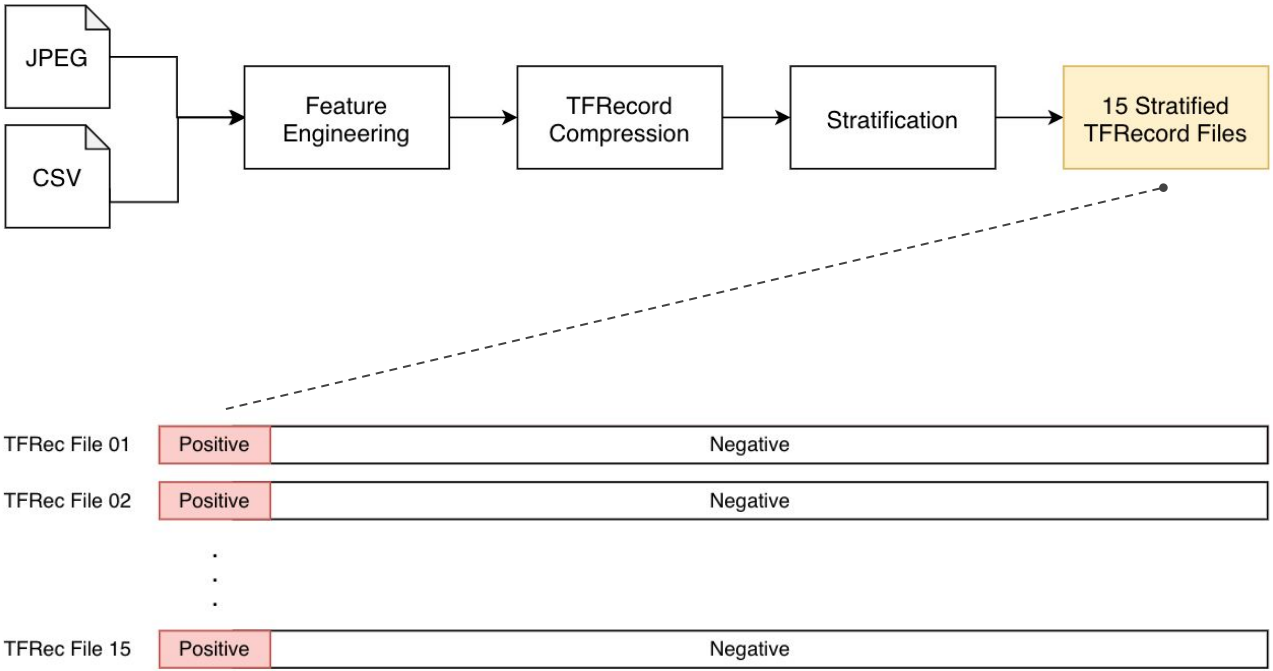
TPU + TFRecord + K-Fold = Efficient Training

Stratified Training Data	EfficientNet	Transfer Learning	Focal Loss

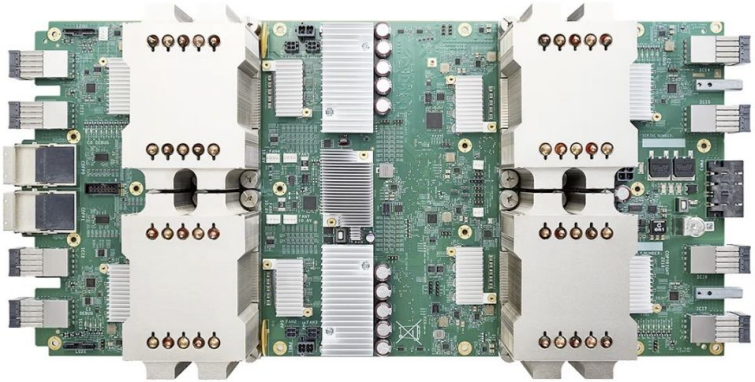
Others

- Additional feature engineering
- Apply random transformation on both training and test datasets
- Restrict image augmentation (e.g. avoid black edges)
- Experiment with hyperparameters using k-fold CV

Stratified Training Data & K-Fold



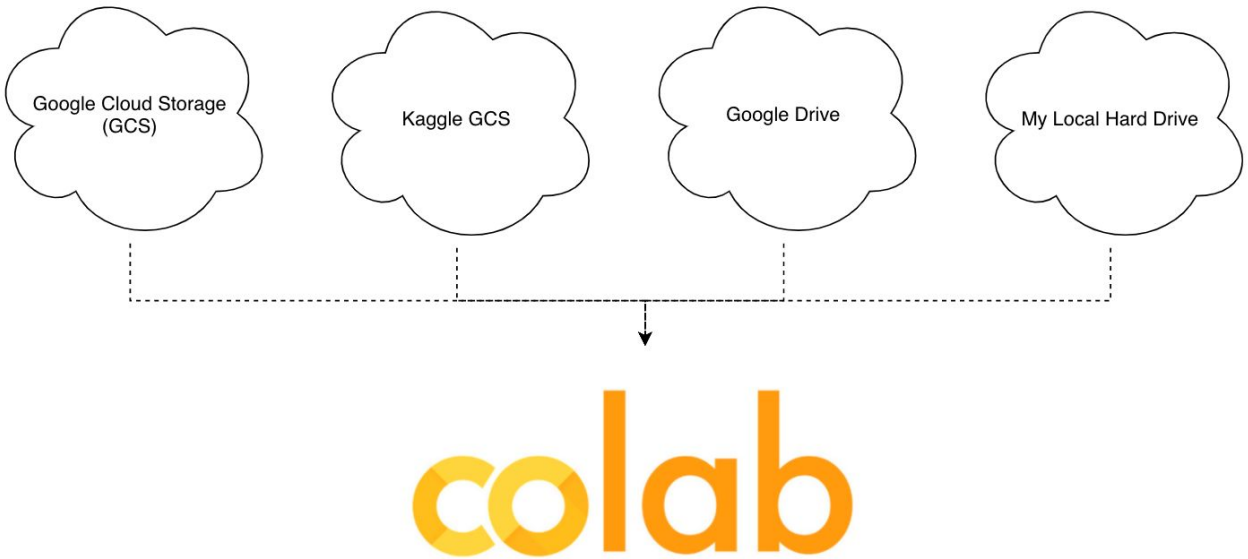
What Is TPU?



colab

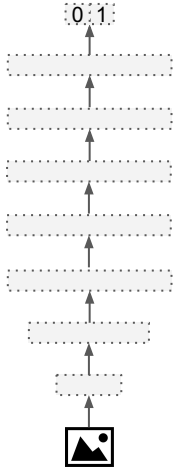
- 4 chips, each has 2 cores
- Processor designed for NN & Tensorflow
- GCS & code changes required (e.g. batch size)!

Colab & TPU

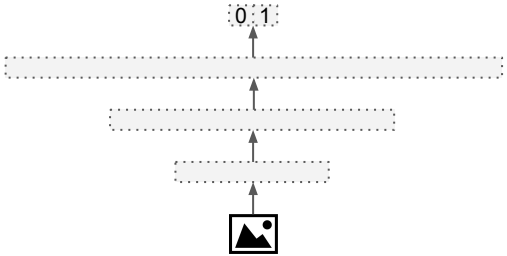

Google Cloud Storage (GCS)
Kaggle GCS
Google Drive
My Local Hard Drive
colab

EfficientNet

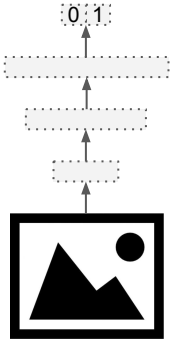
Problem: How do we scale up a CNN?



Deeper



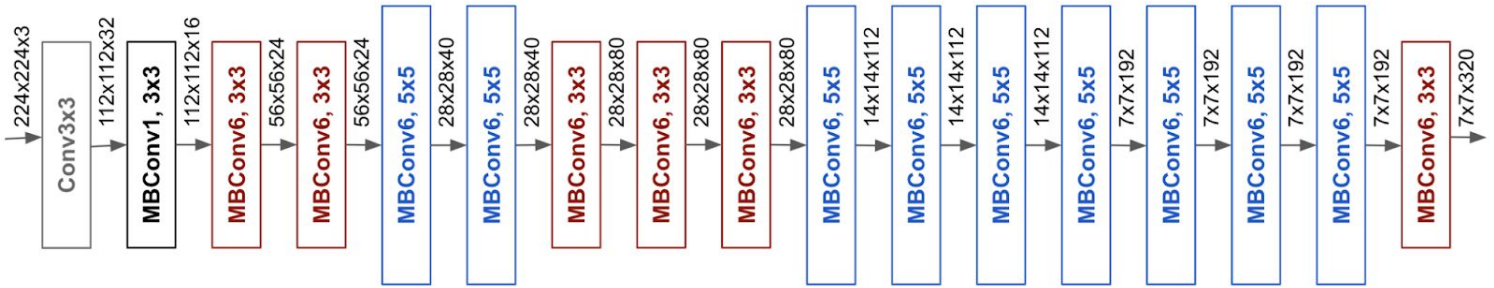
Wider



Higher Resolution

EfficientNet

- Baseline Network
  - EfficientNet-B0 is constructed using neural architecture search, which optimizes both accuracy and efficiency (FLOPS).
- Compound Scaling
  - Grid search to optimally scale up EfficientNet-B0 under a fixed resource constraint
  - Result: B0, B1, B2, ..., B7



EfficientNet-B0

Transferred Learning

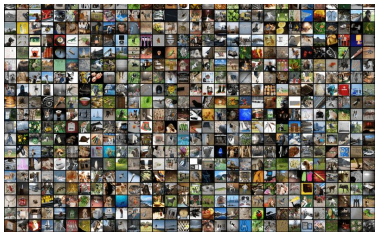
EfficientNet
(Trained by Google)

Transfer Weights

EfficientNet
(My Model)

PRE-TRAINED

IMAGENET



TRAINING CAT PICTURES

kaggle

Focal Loss: ~~Finding~~ Learning Waldo

What if... for every positive example, you have 100+ negative examples?

Solution: Weight Adjustment

1 Heavy-Weight Waldo

False Negative

100+ Small Strangers

Correct Predictions

What Is Focal Loss?

$$FL = -\sum_{i=1}^{C=2} (1 - s_i)^{\gamma} t_i \log(s_i)$$

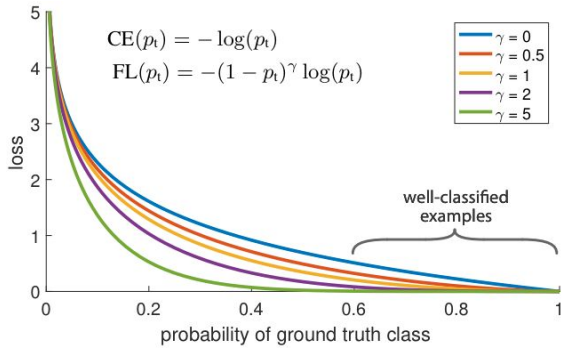
C is the class being predicted
Si is the probability predicted by the neural network

Hyperparameters

- t_i (α on FB's paper) is the probability for positive case
  - 1 in the ideal world
  - Can be adjusted to offset the imbalance in the dataset
- γ is used to adjust the loss curve
  - 2 is the proposed default by FB

Binary Cross-Entropy

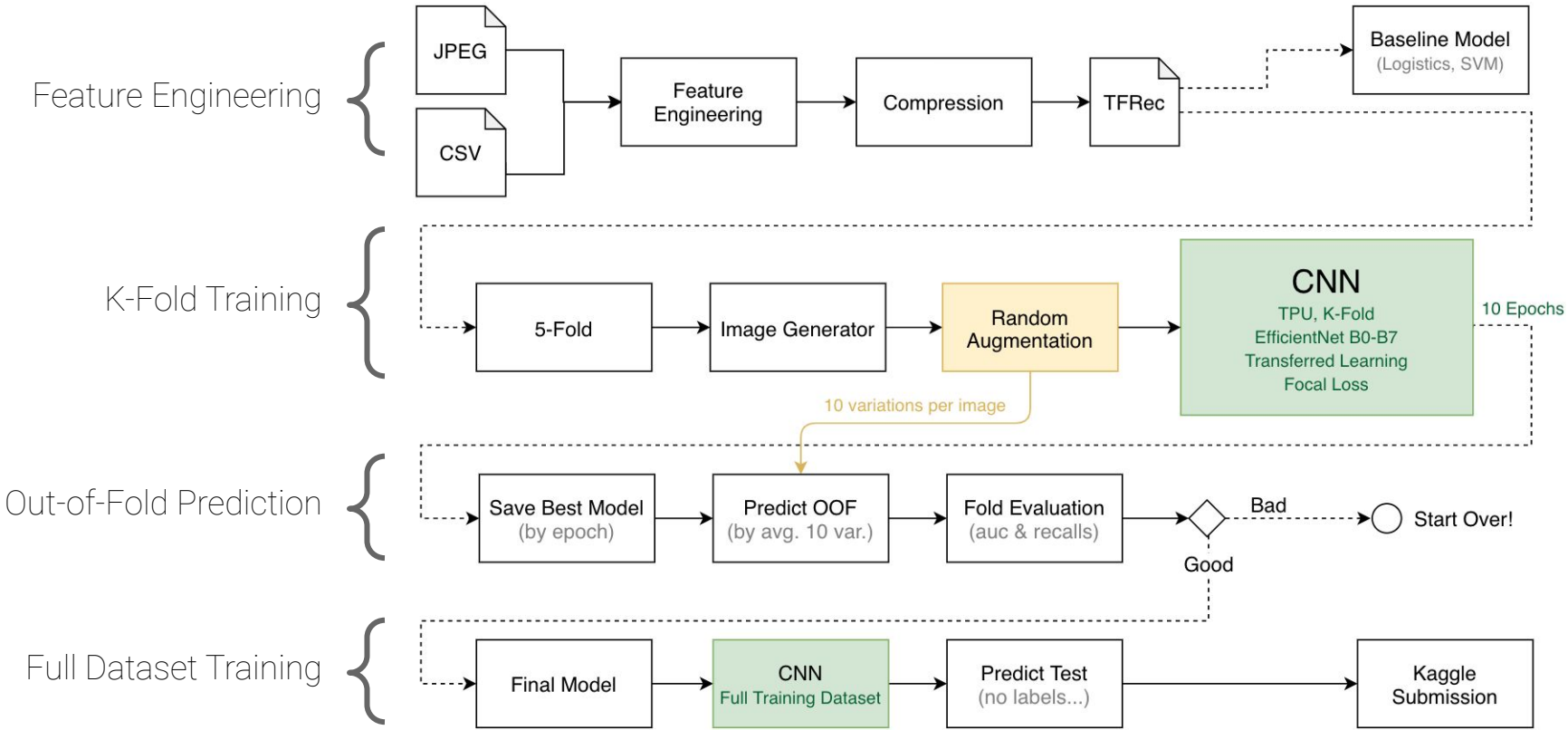
$$CE = -\sum_{i=1}^{C=2} t_i \log(s_i)$$



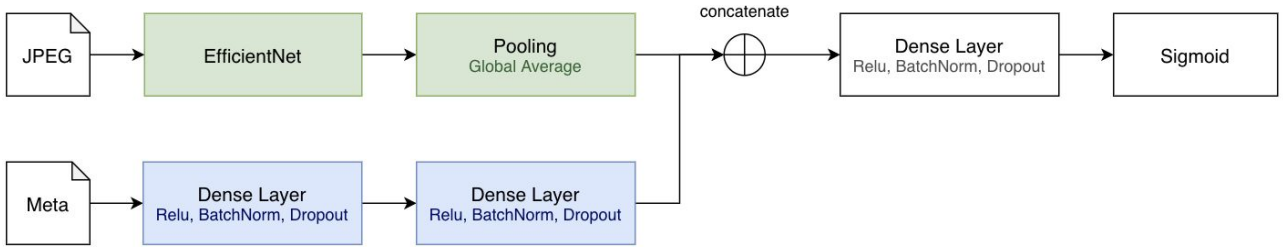
$$L = -\alpha_t (1 - p_t)^{\gamma} \log(p_t)$$

$$\alpha_t = \begin{cases} \alpha & \text{if } y = 1 \\ 1 - \alpha & \text{otherwise} \end{cases}$$

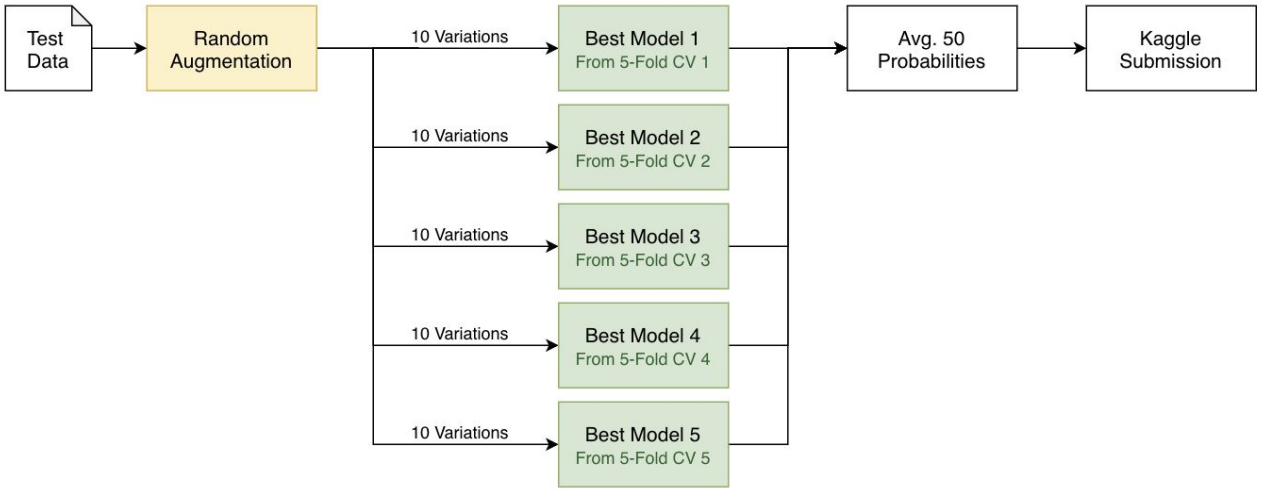
V2 Architecture



V3 Architecture



V3 Predictions



Results

V2 Results

Rank #1, 2019 Competition

Configuration	Sensitivity	AUC
SENet154 SS 224×224	67.2 ± 0.8	94.1 ± 0.4
EN B0 SS 224×224 $C = 8$	66.7 ± 1.8	94.0 ± 0.3
EN B0 SS 224×224	65.8 ± 1.7	92.9 ± 0.8
EN B0 RR 224×224	67.0 ± 1.6	93.3 ± 0.5
EN B1 SS 240×240	65.9 ± 1.6	93.8 ± 0.3
EN B1 RR 240×240	66.8 ± 1.5	93.7 ± 0.7
EN B2 SS 260×260	67.2 ± 1.4	94.1 ± 0.6
EN B2 RR 260×260	67.6 ± 2.0	93.9 ± 0.7
EN B3 SS 300×300	67.8 ± 2.0	94.3 ± 0.5
EN B3 RR 300×300	67.0 ± 1.5	94.0 ± 0.6
EN B4 SS 380×380	67.8 ± 1.1	94.1 ± 0.5
EN B4 RR 380×380	68.1 ± 1.6	94.2 ± 0.5
EN B5 SS 456×456	68.2 ± 0.9	93.5 ± 0.7
EN B5 RR 456×456	68.0 ± 2.2	94.0 ± 0.7
EN B6 SS 456×456	68.8 ± 0.7	93.8 ± 0.5
Ensemble Average	71.7 ± 1.7	95.4 ± 0.5
Ensemble Optimal	$\mathbf{72.5 \pm 1.7}$	$\mathbf{95.4 \pm 0.5}$

Our Results

AUC: 89.05

Problem: Underfitting (?)

- Only 10 epochs of training
- Training and validation show similar performance
- Test Set Performance (Leaderboard):
  - 0.89 AUC
  - Rank: 1,498 (out of 2,600+ entries)

Specs:

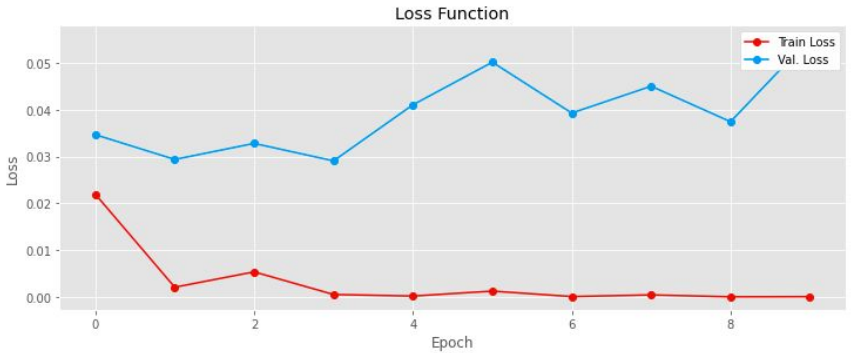
- Model: EfficientNet B7
- Loss Function: Focal Loss
- Transferred Learning: ImageNet
- Epochs: 8
- Batch Size: 64
- Image Size: 256x256
- Random Image Augmentation: True

V3 Results

Problem: Overfitting (high variance), large performance gap between training and validation

Epoch 5:

	Training Data	Validation Data
AUC	0.96	0.71
Recall	0.93	0.15
Precision	0.83	0.36



Demo

What Else Can Be Done?

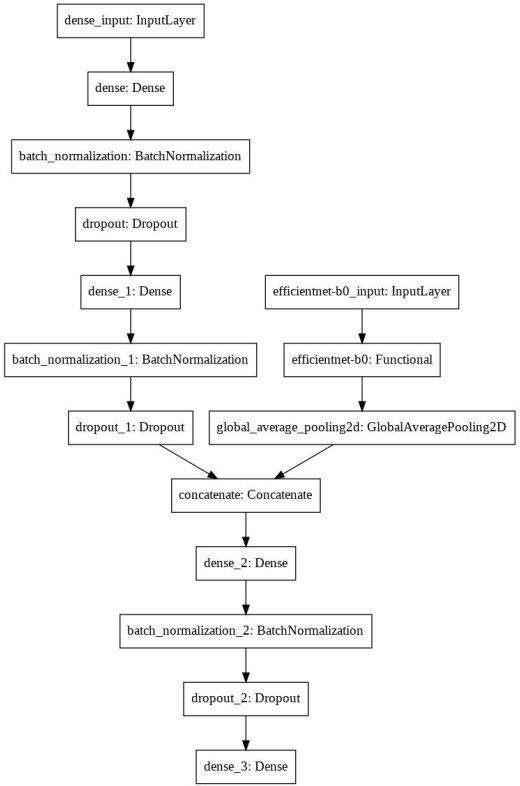
v2 (underfitting) and v3 (overfitting) suffer from different issues and require different treatments.

Other Things to Try:

- Ensemble!
- Hyperparameters Tuning
  - Gradient Clipping
  - Learning Rate Decay
  - Grid Search
- Feature Engineering
  - Hair Simulation
  - Microscope Vignette Simulation
  - Higher Resolutions

Appendix

V3 TensorFlow Model



References

EfficientNet: Improving Accuracy and Efficiency through AutoML and Model Scaling
https://tinyurl.com/y6a78p98

What makes TPUs fine-tuned for deep learning?
https://tinyurl.com/yxqw5x4e

Understanding Categorical Cross-Entropy Loss, Binary Cross-Entropy Loss, Softmax Loss, Logistic Loss, Focal Loss and all those confusing names
https://tinyurl.com/y6x5n9w5

Focal Loss for Dense Object Detection
https://tinyurl.com/y2xua29r