Master in
Computer Vision
Barcelona

M3 – Machine Learning for Computer Vision

Project: Deep learning classification - **Final Presentation**

Group 7: Guillem Capellera, Johnny Nuñez and Anna Oliveras

Outline

- Week 1: Bag of Visual Words framework
- Week 2: Beyond BoVW → SVMs, Spatial Pyramids, Fisher Vectors

Handcrafted methods: Bag of Visual Words

- Week 3: From hand crafted to learnt features
- Week 4: Fine tuning of pre-trained CNNs → Densenet121
- Week 5: Training a CNN from scratch
  - Initial CNN
  - CNN Refinement
  - Residual Connections
  - Residual Network
  - Recap

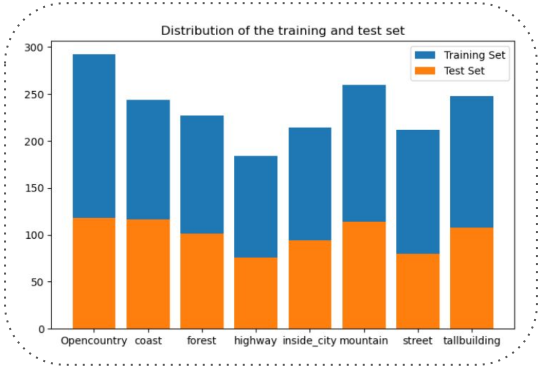
Data driven methods: Deep Convolutional Networks

- Conclusions

Datasets

- We have 8 classes: coast, forest, highway, inside city, mountain, open country, street, tall buildings

- Big dataset : MIT_split → total of 2288 images



- Small dataset: MIT_small_train_1 → Train with only 50 images for each class !

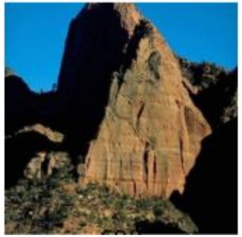
Opencountry

tallbuilding

highway

street

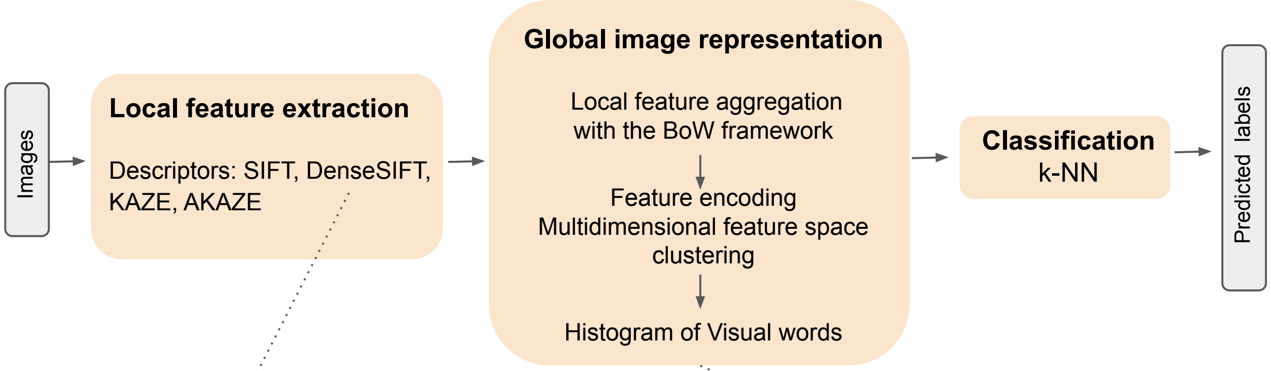
mountain



coast

inside_city

Week 1: Bag of Visual Words framework


Images
Local feature extraction
Descriptors: SIFT, DenseSIFT, KAZE, AKAZE
Global image representation
Local feature aggregation with the BoW framework
Feature encoding
Multidimensional feature space clustering
Histogram of Visual words
Classification
k-NN
Predicted labels

Grid search best parameters (Optuna)

Descriptor: DenseSIFT with

- n_features = 251
- patch_size = 3
- step size = 75

Clustering

- Codebook size k = 1024

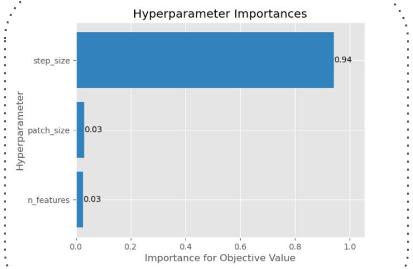
Dimensionality reduction

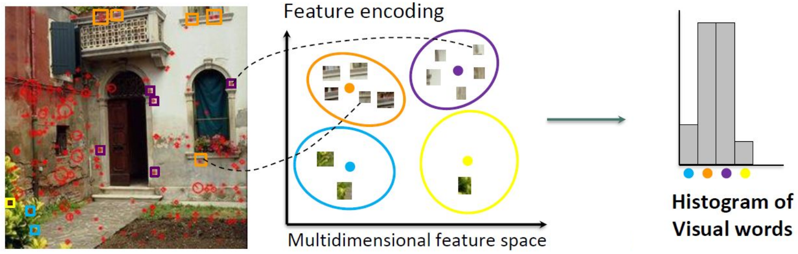
- PCA with n_components = 46

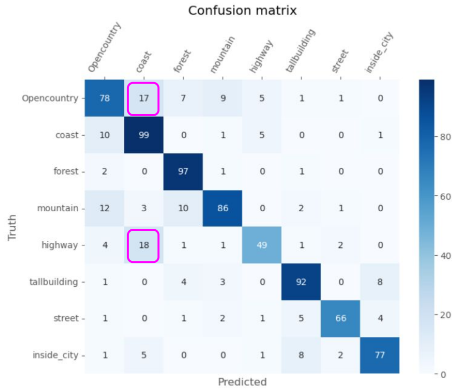
Classifier

- n_neighbors = 18
- distance = euclidean

Accuracy = 0.95 **f1-score** = 0.8





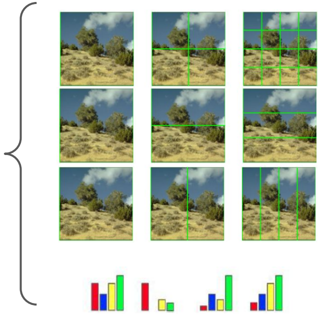


Truth \ Predicted	Opencountry	coast	forest	mountain	highway	tallbuilding	street	inside_city
Opencountry	78	17	7	9	5	1	1	0
coast	10	99	0	1	5	0	0	1
forest	2	0	97	1	0	1	0	0
mountain	12	3	10	86	0	2	1	0
highway	4	18	1	1	49	1	2	0
tallbuilding	1	0	4	3	0	92	0	8
street	1	0	1	2	1	5	66	4
inside_city	1	5	0	0	1	8	2	77

Week 2: Beyond BoVW

Spatial Pyramids

Multiple divisions:
1x1, 2x2, 4x4
vertical 1x2, 1x3, 1x4
horizontal 2x1, 3x1, 4x1



SVM and Kernels

- Linear
- RBF
- Histogram intersection

Feature normalization

- Power Norm
- L2 Norm

Fischer Vectors

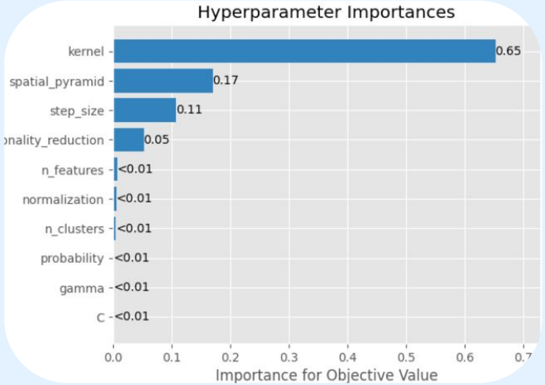
Inlude higher order statistics: mean, covariance of local descriptors → GMM clustering

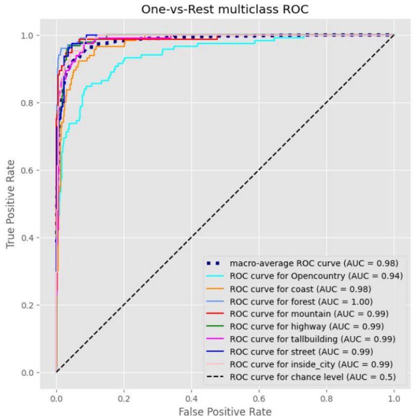
Accuracy = 0.96 (Without hyperparameter optimization)
F1 score = 0.83

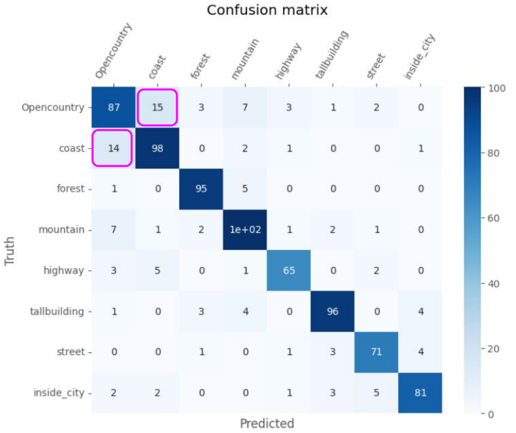
Grid search best parameters (Optuna)→ We use Dense SIFT

- num features = 178
- **step size = 18**
- num cluster = 798
- num components = 69
- gamma = 0.00445
- C = 4.38
- **dim_reduc = PCA**
- **kernel = RBF**
- Normalization = power
- **Spatial_pyramid = vertical 1x4**

Accuracy = 0.96
F1 score = 0.86



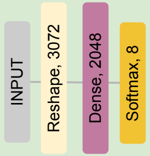



Confusion matrix

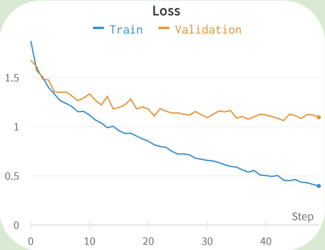
Truth \ Predicted	Opencountry	coast	forest	mountain	highway	tallbuilding	street	inside_city
Opencountry	87	15	3	7	3	1	2	0
coast	14	98	0	2	1	0	0	1
forest	1	0	95	5	0	0	0	0
mountain	7	1	2	1e+02	1	2	1	0
highway	3	5	0	1	65	0	2	0
tallbuilding	1	0	3	4	0	96	0	4
street	0	0	1	0	1	3	71	4
inside_city	2	2	0	0	1	3	5	81

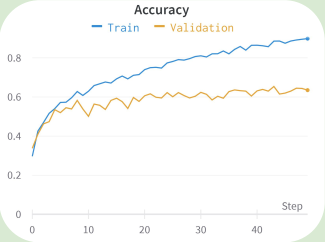
Week 3: From hand crafted to learnt features

Initial Model

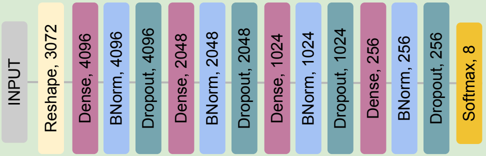


- 6M parameters
- Accuracy = **0.61**

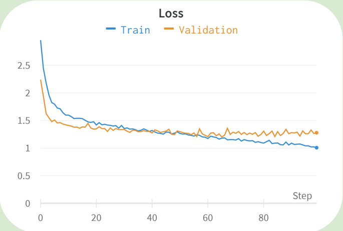


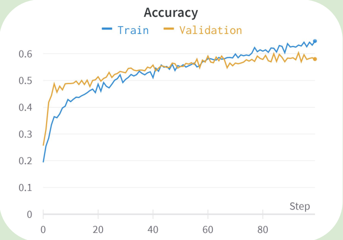


Best Model



- 23M parameters
- Accuracy = **0.596**

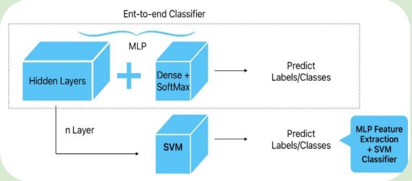




Deep features + SVM

Extracting the features after each dense layer output → similar results (best with output dense 4096)

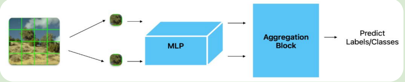
- Accuracy = **0.41**



Patch based MLP

→ Best patch size: 32 x 32
→ Best aggregation: mean

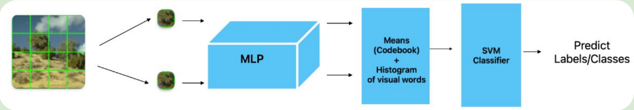
- Accuracy = **0.77**



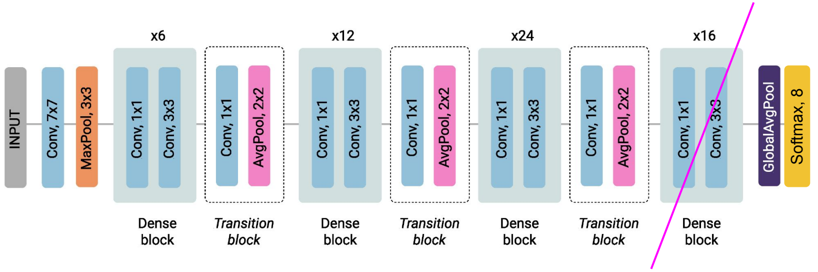
Patch based deep features + BoVW

→ Num clusters too low —> too much generalization of the features
→ Num clusters too big → too much specificity of the features
→ Best number of clusters (codebook size) = 256

- Accuracy = **0.72**



Week 4: Fine Tuning DenseNet121[1]



29% less parameters than the original model

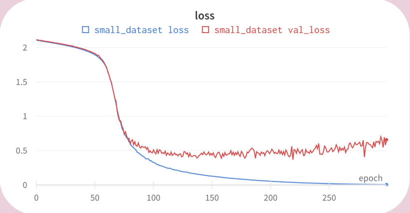
Making the network smaller

MIT_split dataset

Model	Epochs	Num parameters	Validation accuracy
Original	300	7M	0.9542
Removing 1 DB	**300**	**5M**	**0.941**
Removing 2 DB	300	1.5M	0.825

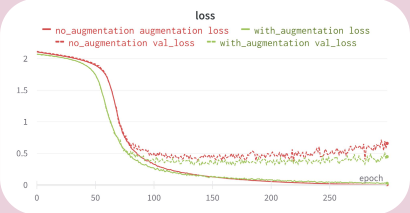
Using MIT_small dataset

- **Overfitting**
- Accuracy drops to **0.845**



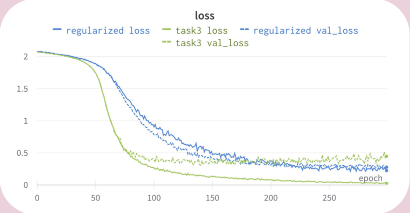
Data augmentation

- Horizontal Flip = True
- Zoom Range = 20%
- Accuracy increases to **0.895**
- Validation loss does not increase



Improve learning curve

- Early Stopper
- Reduce LR
- BatchNormalization and Dropout
  → Accuracy increases to **0.915**

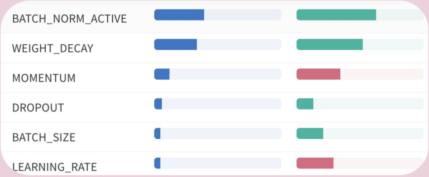


Hyperparameter Optimization

→ Accuracy increases to **0.9518**

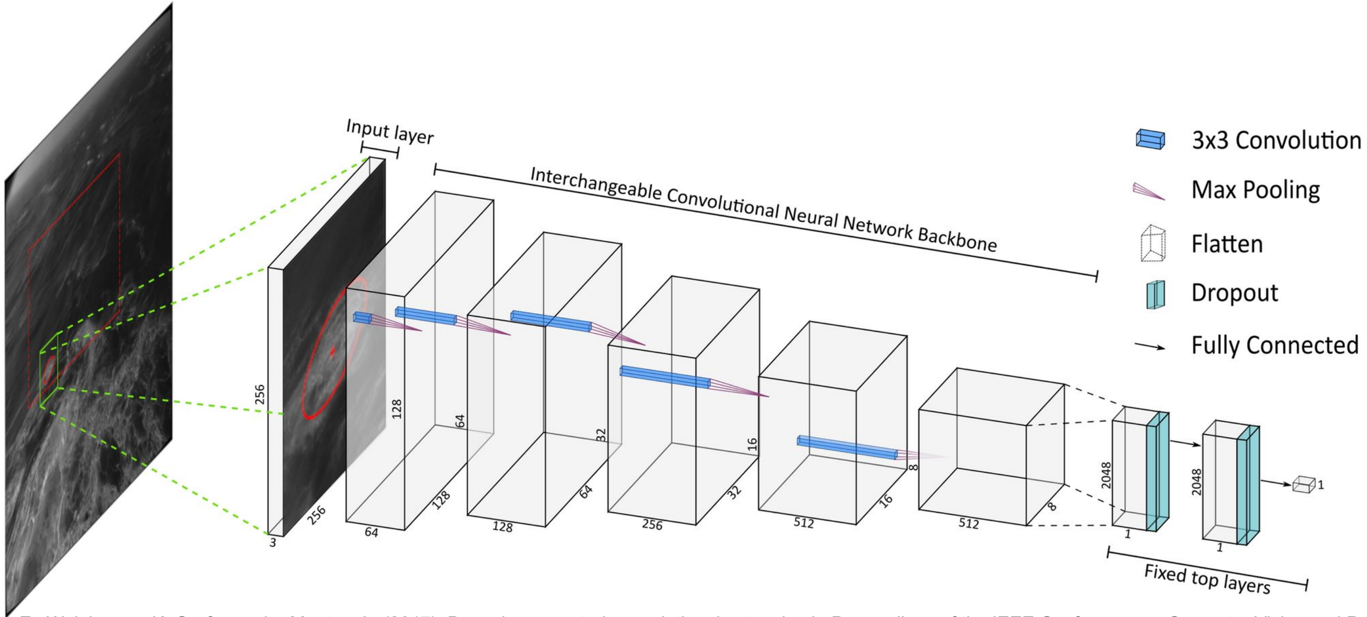
Optimizer = Adamax
LR = 0.0001
Dropout = 0.5

Wight decay = 0.3
Batch_ size = 10
BatchNormalization



[1] Huang, G., Liu, Z., Weinberger, K. Q., & van der Maaten, L. (2017). Densely connected convolutional networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (pp. 4700-4708). https://arxiv.org/abs/1608.06993

Building a CNN from scratch

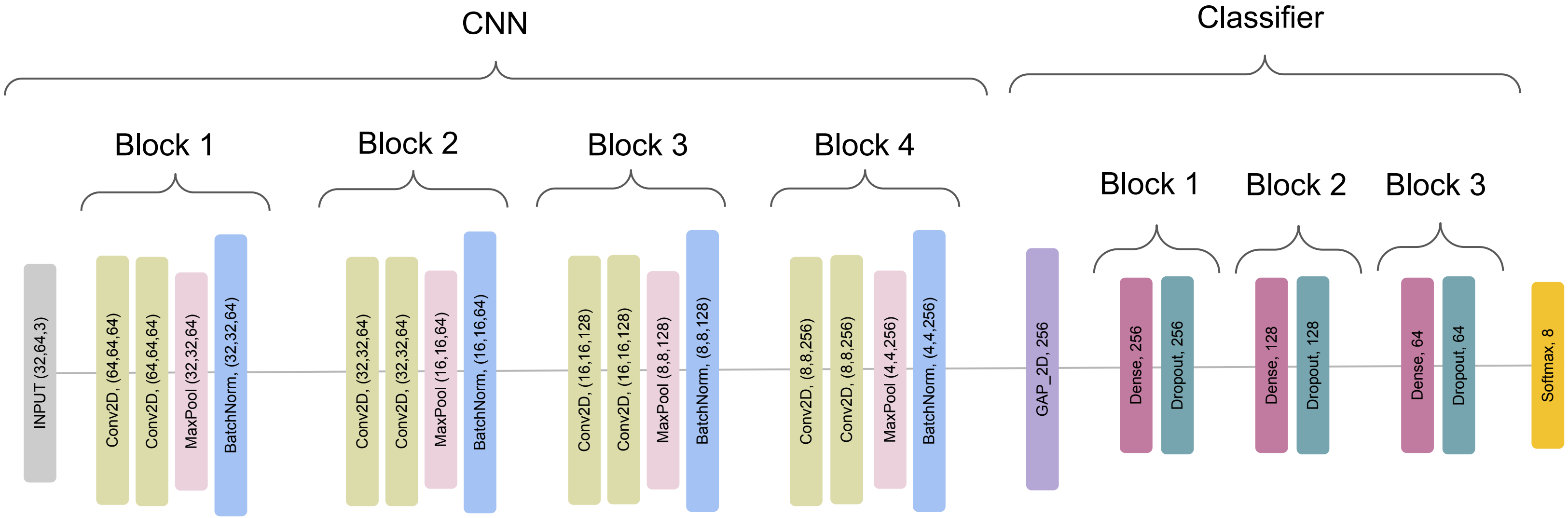


[1] Huang, G., Liu, Z., Weinberger, K. Q., & van der Maaten, L. (2017). Densely connected convolutional networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (pp. 4700-4708). https://arxiv.org/abs/1608.06993

Manual search of the CNN network

Params	Accuracy
1.5M	0.75

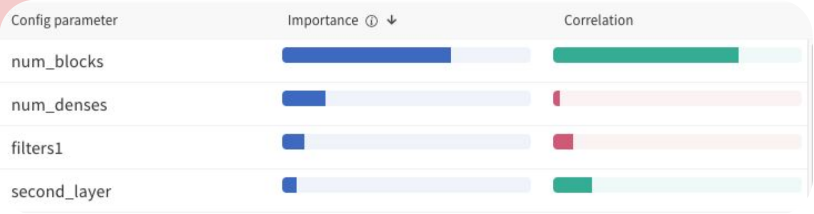
- CNN
  - 4 blocks → Conv2D + Conv2D + MaxPool + BatchNorm
- Classifier
  - Global Average Pooling
  - 3 blocks → Dense + Dropout



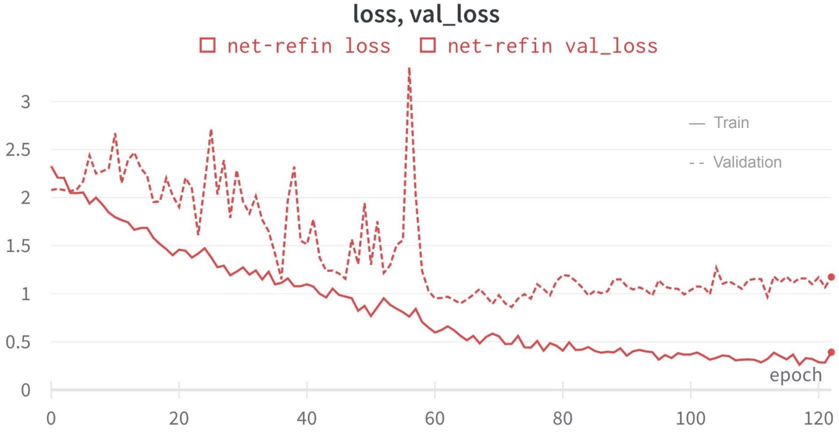
Network refinement

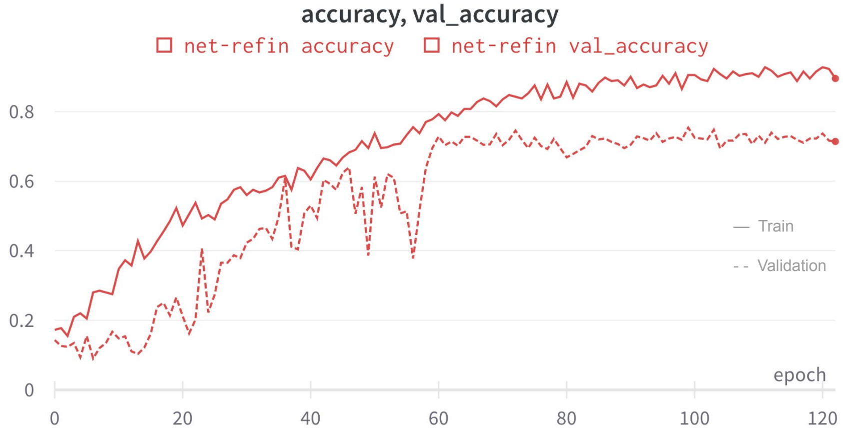
Params	Accuracy
1.3M ↓	0.75

- Number of CNN blocks → 2, 3, **4**, 5
- For each CNN block → Add a second layer? **True**, False
- **Dimensionality first filter** → **32**, 64
- Number of dense layers → 1, 2, **3**, 4



Config parameter	Importance ⓘ ↓	Correlation
num_blocks		
num_denses		
filters1		
second_layer		

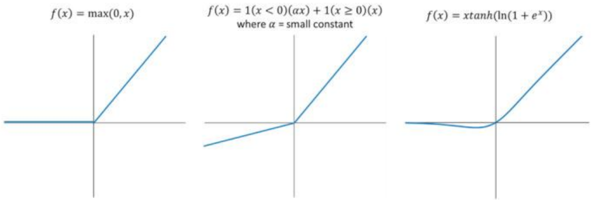


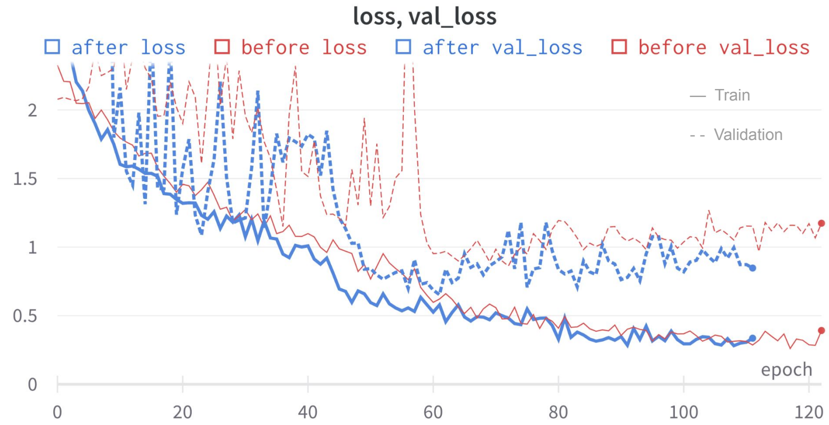


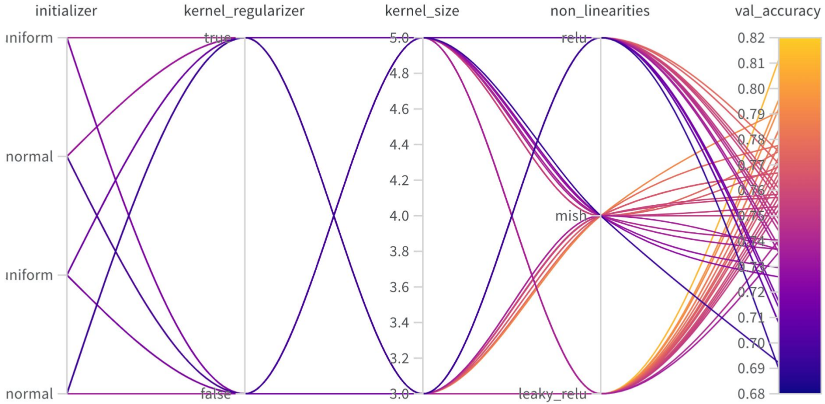
Initializer & activation function

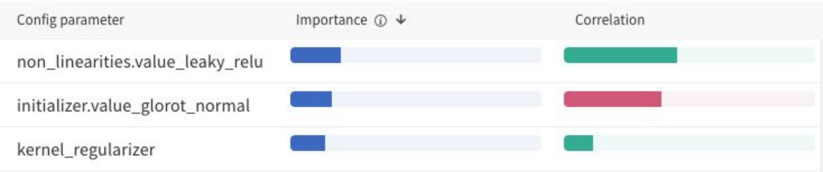
Params	Accuracy
1.3M	0.81 ↑

- Initializer → GlorotUniform, GlorotNormal, HeUniform, HeNormal
- Activation → Relu, LeakyRelu, Mish
- Kernel size → 3x3 or 5x5
- Kernel regularizer → True, False







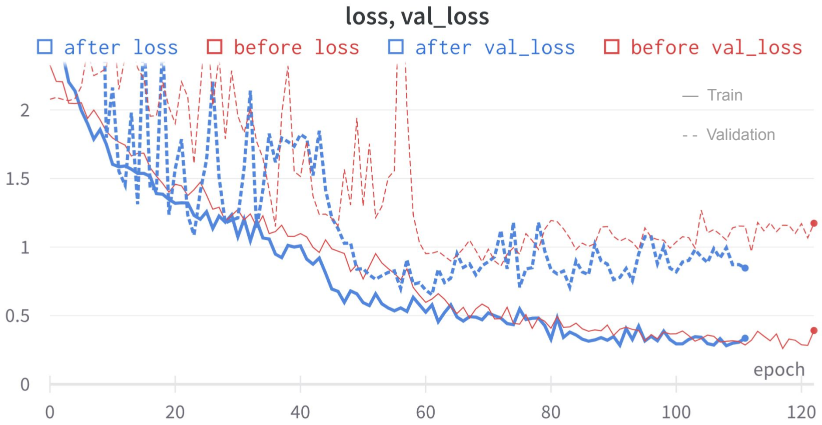


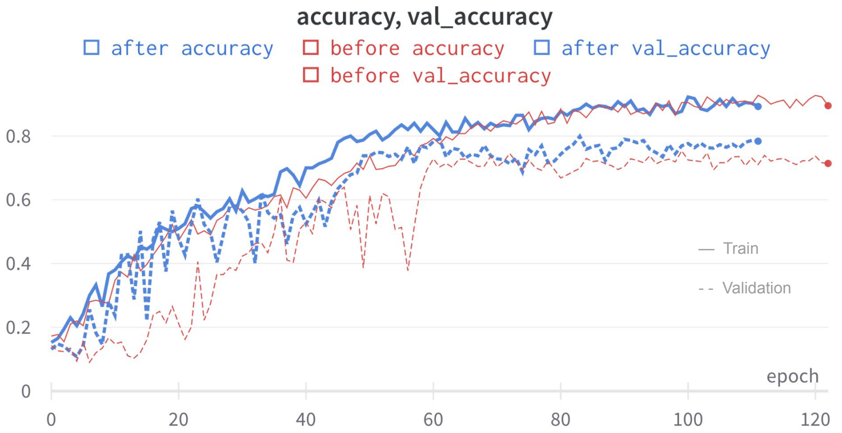
Config parameter	Importance ⓘ ↓	Correlation
non_linearities.value_leaky_relu		
initializer.value_glorot_normal		
kernel_regularizer		

Initializer & activation function

Params	Accuracy
1.3M	0.81 ↑

- Initializer → GlorotUniform, GlorotNormal, HeUniform, HeNormal
- Activation → Relu, LeakyRelu, Mish
- Kernel size → 3x3 or 5x5
- Kernel regularizer → True, False

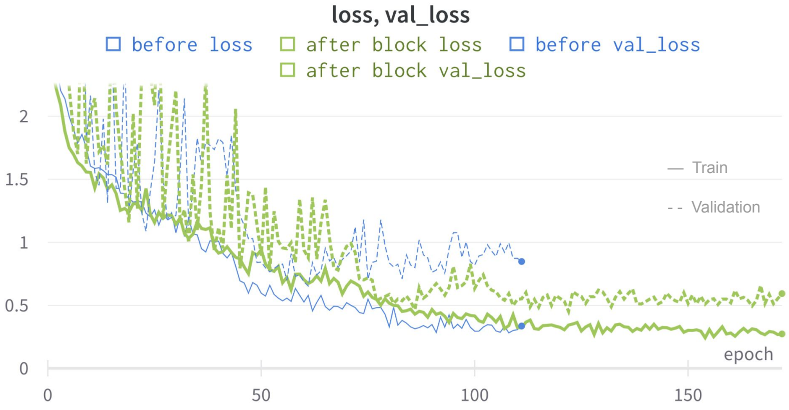


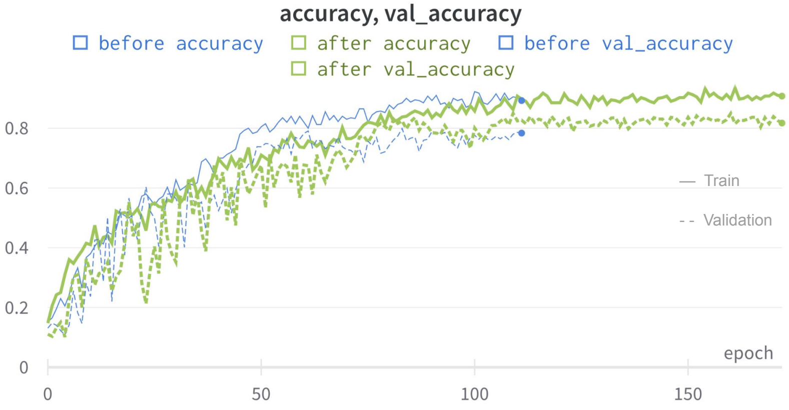


Hyperparameter optimization

Params	Accuracy
1.3M	0.85 ↑

Sweep	Values	Best	Sensitivity
LR	[0.01, 0.005, 0.001]	0.001	High
Batch Size	[8, 16, 32, 64]	32	Medium
Optimizer	[Adam, SGD]	Adam	Low
Dropout values	[0.3 , 0.4, 0.6]	0.4	Medium
Horizontal Flip	[True, False]	True	High
Rotation	[0, 15]	0	Medium
Width Shift	[0, 0.1]	0.1	Medium
Height Shift	[0, 0.1]	0.1	Medium
Shear Range	[0, 0.1]	0	Medium
Zoom Range	[0, 0.1]	0	High





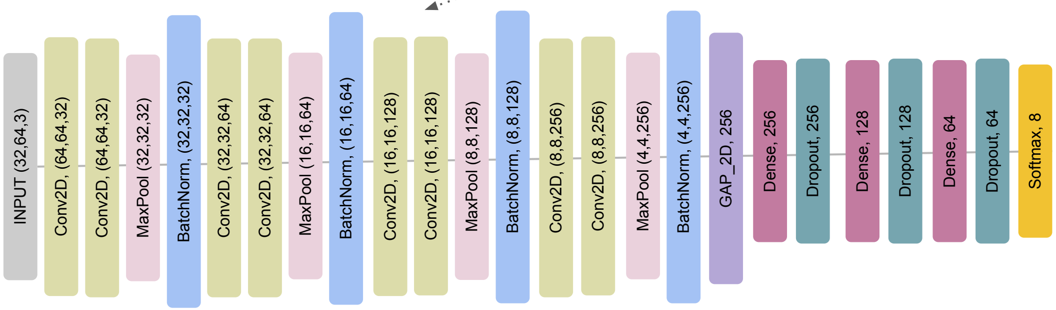
Parameter refinement

- Reduce the number of **filters** at each CNN block

Params	Accuracy
1.3M	0.85

Params	Accuracy
380k ↓↓	0.80 ↓

CNN blocks	Before	After
1st block	32, 32	32, 32
2nd block	64, 64	**32, 32**
3th block	128, 128	**64, 64**
4th block	256, 256	**128, 128**



INPUT (32,64,3)

Conv2D, (64,64,32)

Conv2D, (64,64,32)

MaxPool (32,32,32)

BatchNorm, (32,32,32)

Conv2D, (32,32,64)

Conv2D, (32,32,64)

MaxPool (16,16,64)

BatchNorm, (16,16,64)

Conv2D, (16,16,128)

Conv2D, (16,16,128)

MaxPool (8,8,128)

BatchNorm, (8,8,128)

Conv2D, (8,8,256)

Conv2D, (8,8,256)

MaxPool (4,4,256)

BatchNorm, (4,4,256)

GAP_2D, 256

Dense, 256

Dropout, 256

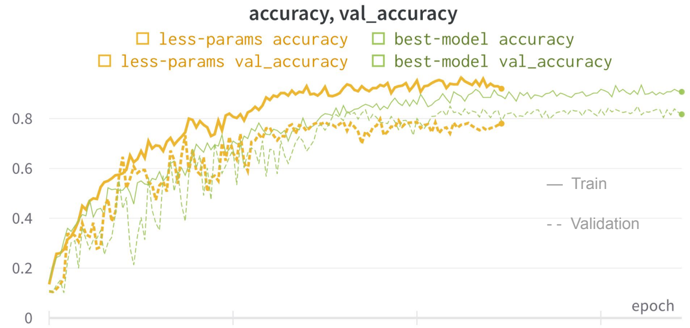
Dense, 128

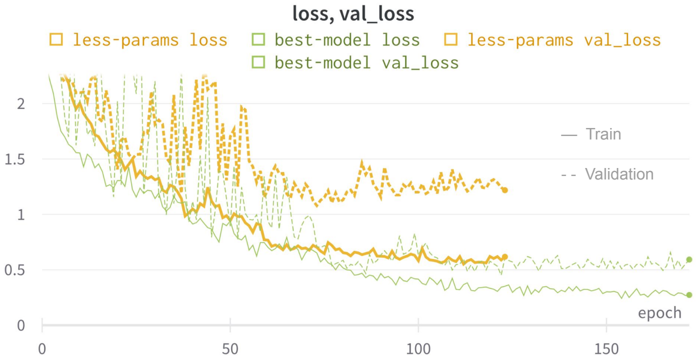
Dropout, 128

Dense, 64

Dropout, 64

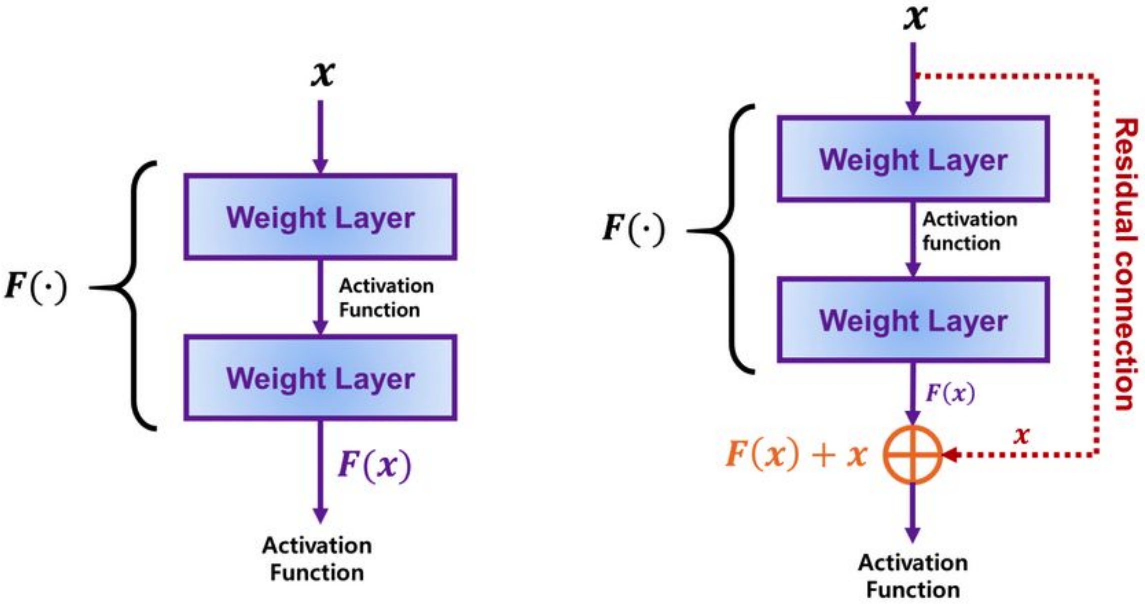
Softmax, 8





Residual Connections

- Residual Connections between blocks
- Reduce dimensionality of the filters



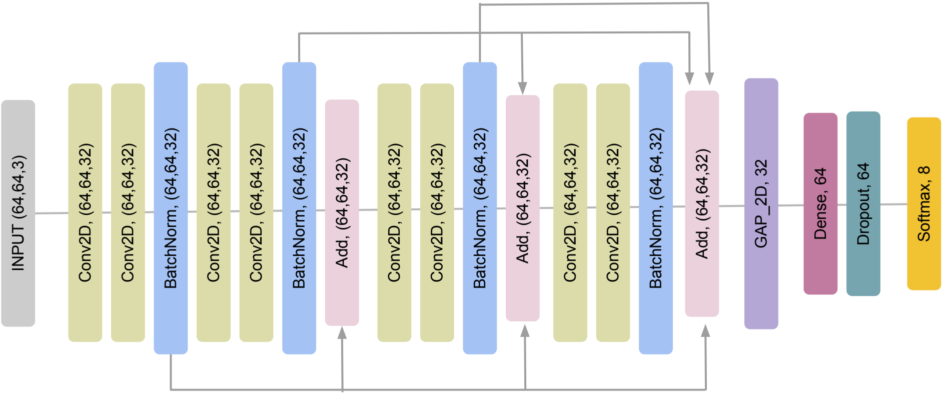
Benefits

- Converge more easily
- More Stability Training
- Better Generalization
- Easy implementation
- Improve accuracy

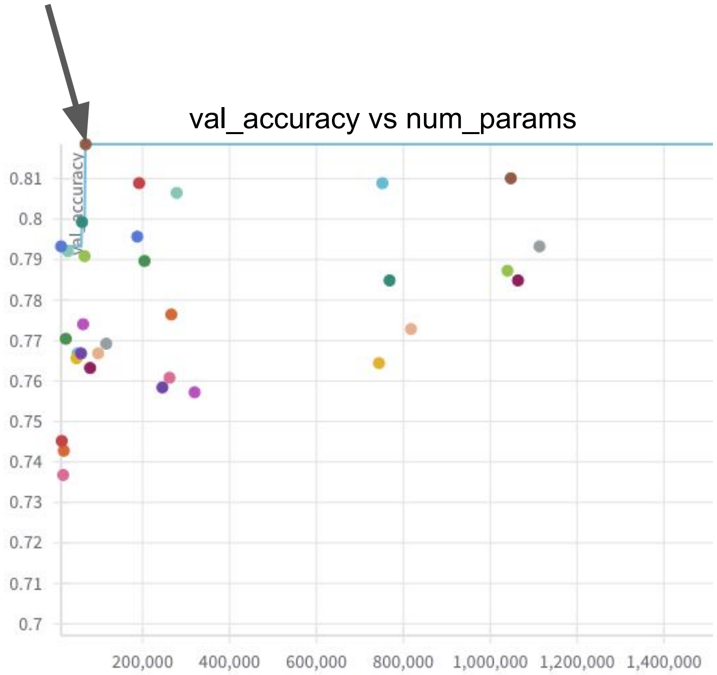
Residual network search

Params	Accuracy
68k	0.82

- CNN blocks → 2, 3, **4**, 5
- Num filters → 16, **32**, 64, 128, 256
- Dense layers → 0, **1**, 2, 3



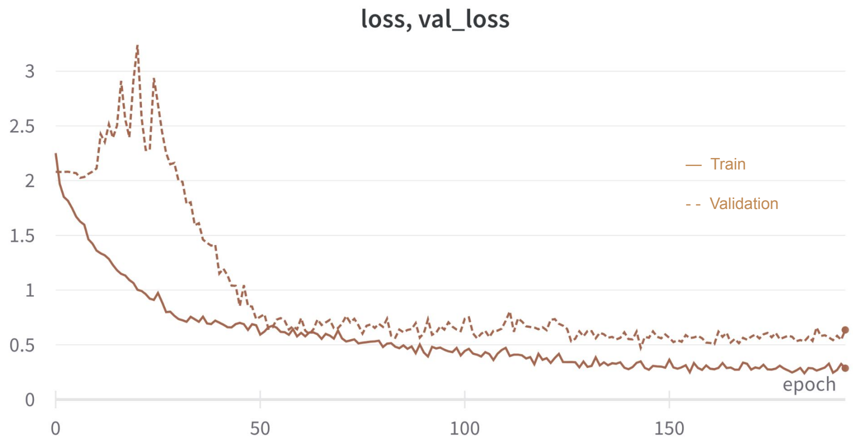
68k!!!

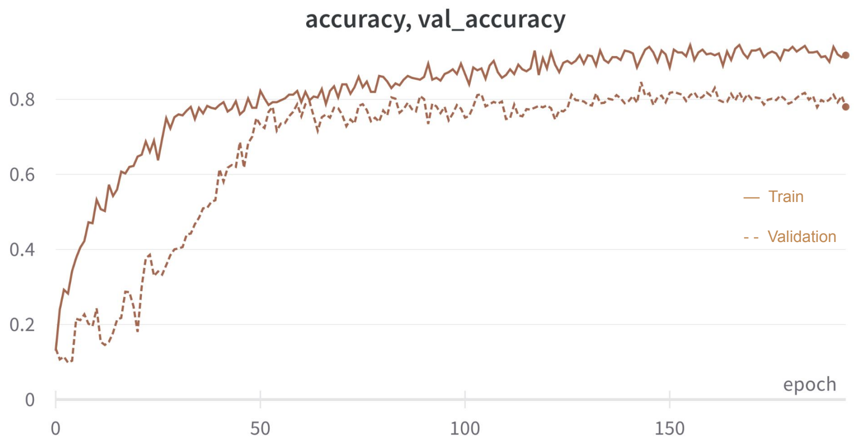


Residual network search

Params	Accuracy
68k	0.82

- CNN blocks → 2, 3, **4**, 5
- Num filters → 16, **32**, 64, 128, 256
- Dense layers → 0, **1**, 2, 3

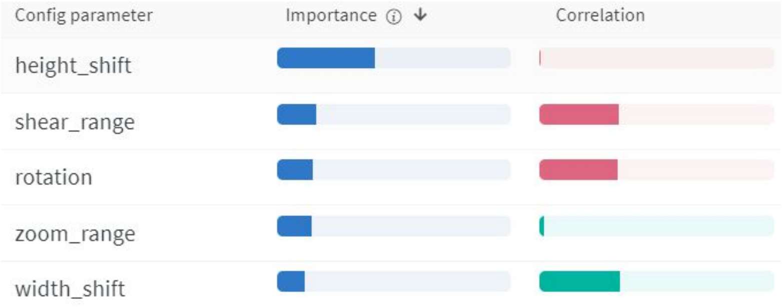

loss, val_loss


accuracy, val_accuracy

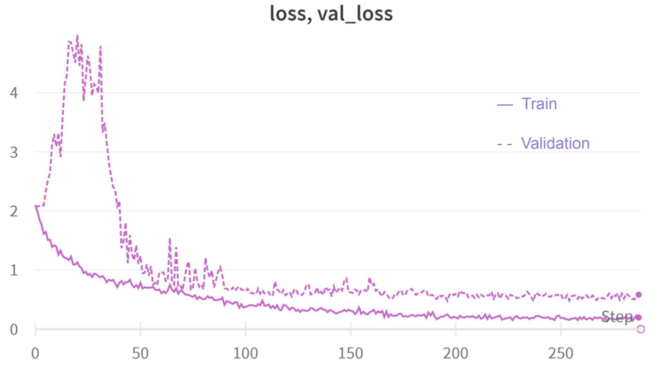
Data augmentation refinement

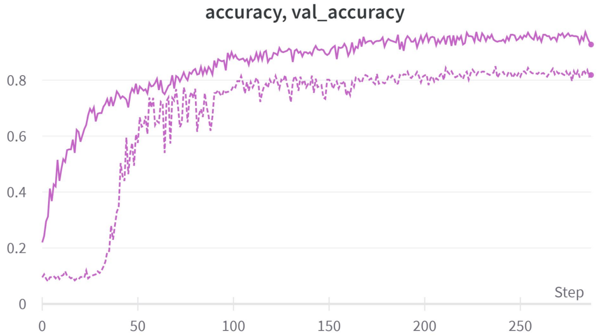
Params	Accuracy
68k	0.85 ↑

- Horizontal Flip → True
- Rotation [0, 15] → **0**
- Width Shift [0, 0.1] → **0.1**
- Height Shift [0, 0.1] → **0**
- Shear Range [0, 0.1] → 0
- Zoom Range [0, 0.1] → 0

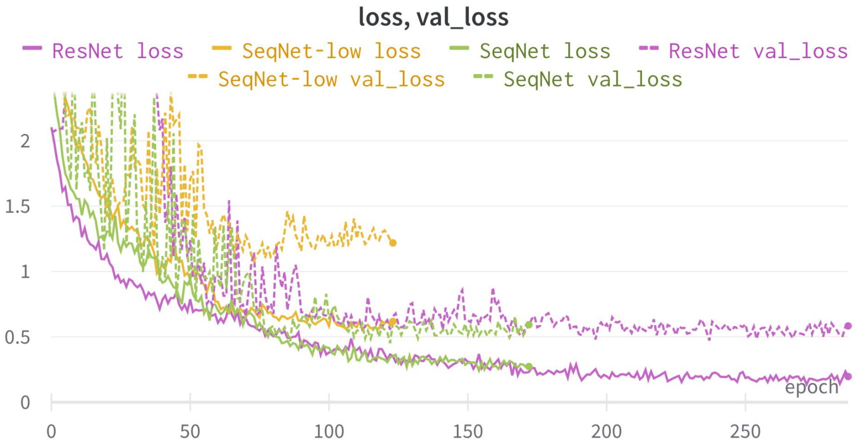


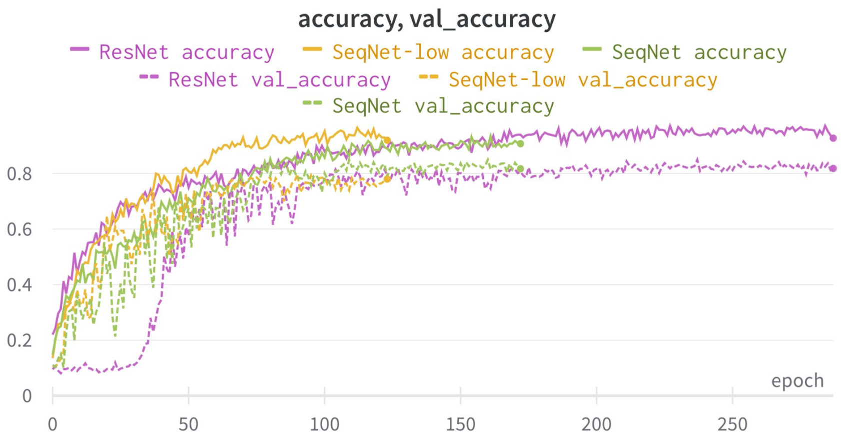
Config parameter	Importance ⓘ ↓	Correlation
height_shift		
shear_range		
rotation		
zoom_range		
width_shift		





Recap week 5





Params	Accuracy
1.3M	0.85

Params	Accuracy
380k	0.80

Params	Accuracy
68k	0.85

Recap week 5

- Good results with BOVW and spatial pyramids (**0.95** accuracy)
- MLP → too simple for our problem in all the cases (**0.77** accuracy)
- Reduced data → challenging and requires the use of **data augmentation** and other techniques to achieve the desired results
- Fine-tuning the DenseNet121 → best results (**0.96** accuracy) and we were able to reduce **30%** of the network parameters to **5M**
- Building a network from scratch is time consuming and difficult to optimize
- Residual connections → help the network to converge easily and generalize.

  → **0.85** accuracy with **68k** parameters
- Our results are limited due to the lack of data