Deep Learning – MAI

Autonomous lab – CNNs

MAMe: Museum Art Medium dataset

Alberto Becerra
Riccardo Corsiglia

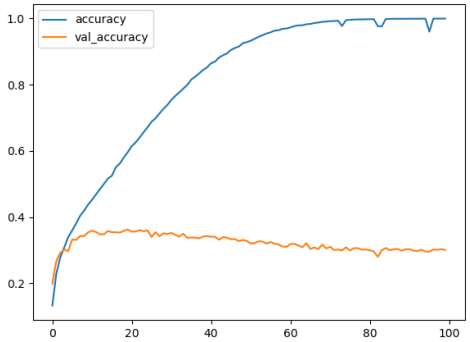
Table of Contents

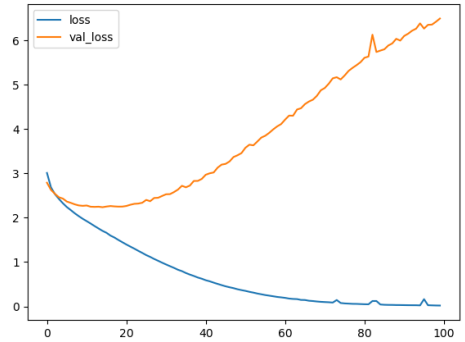
#1: Overfit to find Difficult Classes

First Training. No previous attempts

- 1 * 3x3 Convolutional kernel (ReLU)
- 1 * 4x4 MaxPool
- 64 units dense layer (ReLU)
- 29 units output layer (SoftMax)

Accuracy Train: 0.97
Accuracy Val: 0.307



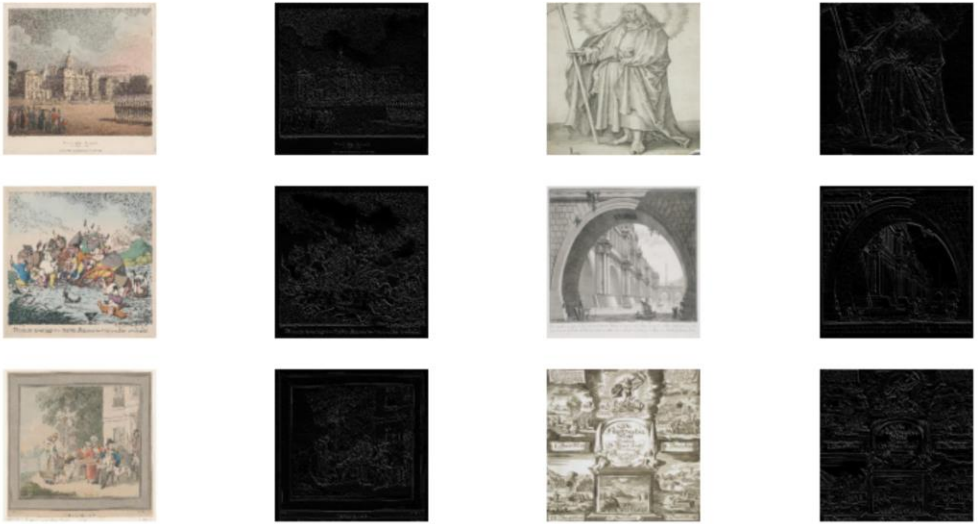


	precision	recall	f1-score	support
Albumen photograph	1.00	1.00	1.00	700
Bronze	1.00	0.97	0.99	700
Ceramic	1.00	0.98	0.99	700
Clay	1.00	0.92	0.96	700
Engraving	0.92	1.00	0.96	700
Etching	1.00	0.79	0.88	700
Faience	1.00	0.99	0.99	700
Glass	0.92	1.00	0.96	700
Gold	0.99	0.97	0.98	700
Graphite	0.93	1.00	0.96	700
Hand-colored engraving	1.00	1.00	1.00	700
Hand-colored etching	1.00	0.85	0.92	700
Iron	0.88	1.00	0.94	700
Ivory	0.99	0.99	0.99	700
Limestone	0.98	1.00	0.99	700
Lithograph	0.96	1.00	0.98	700
Marble	1.00	1.00	1.00	700
Oil on canvas	1.00	1.00	1.00	700
Pen and brown ink	0.96	1.00	0.98	700
Polychromed wood	0.99	1.00	1.00	700
Porcelain	0.99	0.98	0.99	700
Silk and metal thread	1.00	0.97	0.99	700
Silver	1.00	0.95	0.97	700
Steel	1.00	1.00	1.00	700
Wood	0.99	0.99	0.99	700
Wood engraving	1.00	0.96	0.98	700
Woodblock	0.93	1.00	0.96	700
Woodcut	0.87	1.00	0.93	700
Woven fabric	1.00	0.94	0.97	700

- **Hand-colored etching** → Engraving, Graphite, Lythograph, Pen and Brown Ink
- Silver, Gold, Bronze, Clay → **Iron**
- Woven Fabric, Wood Engraving, Hand Colored Etching, Etching → **Woodcut**

Label Gold VS Iron

Label Hand-colored etching VS Engraving



#2: Increase Complexity: More Filters

In #1 really basic feature extraction was done. Using 1 convolutional unit led to edge detector And led to fast overfitting

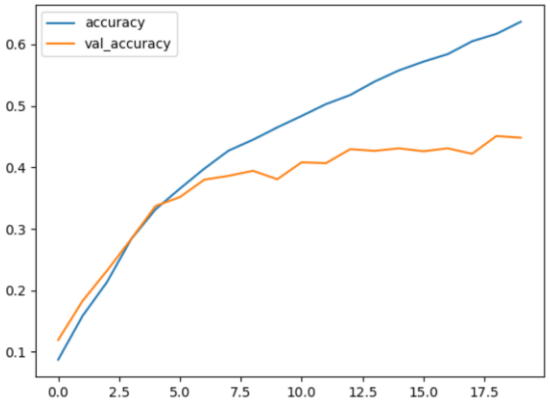
In #2, more convolutional kernels are added to the layer so that different perspectives of the same object can be used for classification

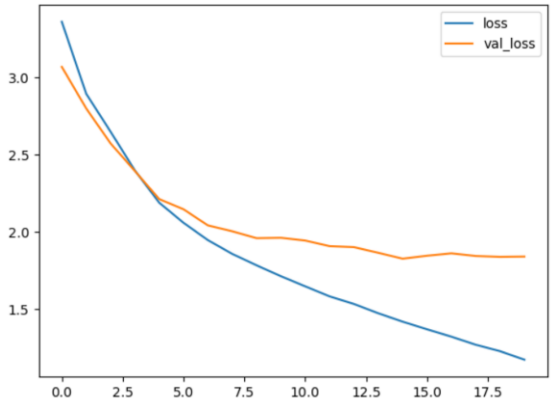
- 3x3 16 units Convolutional kernel (ReLU)
- 4x4 MaxPool
- 64 units dense layer (ReLU)
- 29 units output layer (SoftMax)

Batch Size: 128
Optimizer: Adam
Learning Rate: 0.0005

Accuracy Train: 0.66
Accuracy Val: 0.45

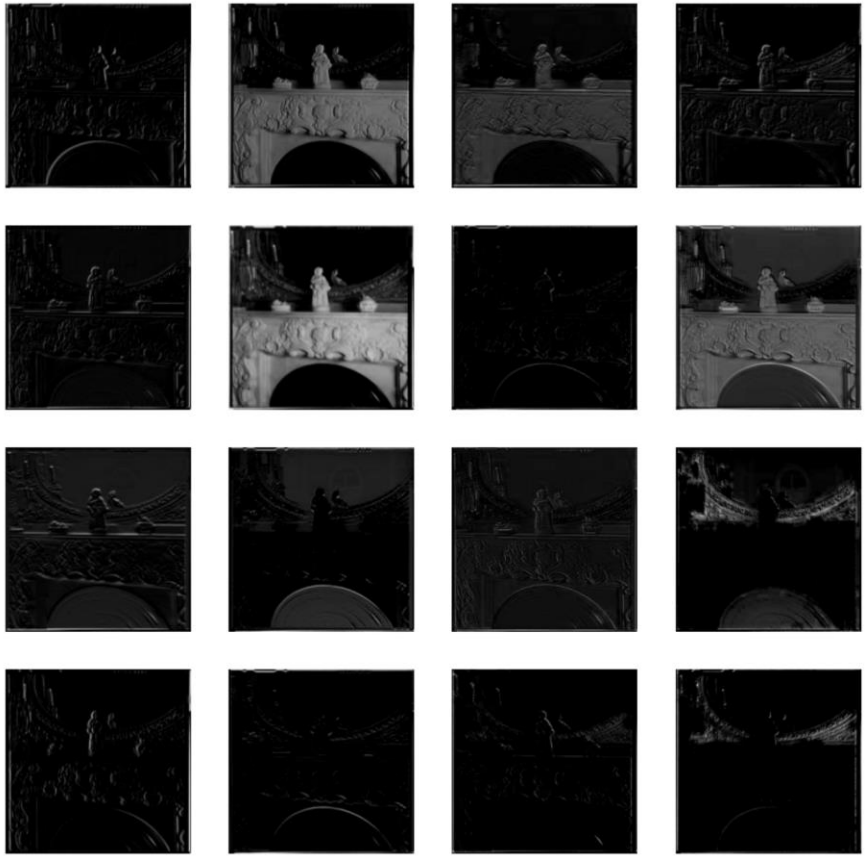
This small change, easily outperforms #1 in Validation. Overfitting starts, approximately, when the validation loss is below 2.0 and accuracy above 0.40





#2: Increase Complexity: More Filters

Marble (predicted: Wood)



#3: Increase Complexity: 2 Conv Layers

In #2, the model is not really understanding The latent feature space but memorising simple Features.

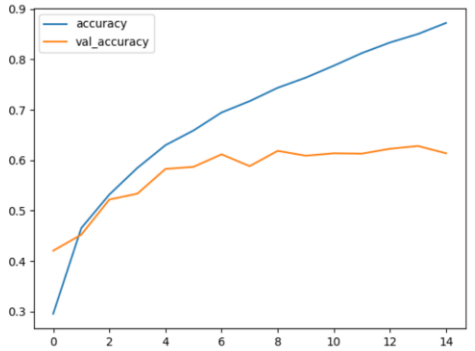
In #3 an extra convolutional layer provides Deeper patterns with more comprised and Concrete image independent information

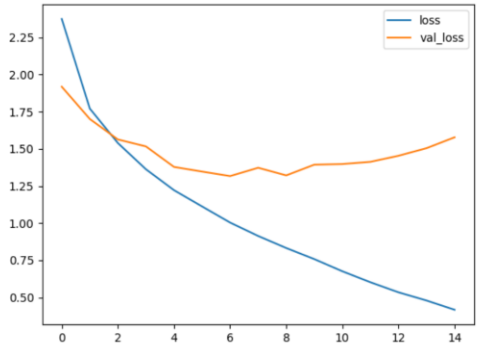
- 3x3 16 units Convolutional kernel (ReLU)
-4x4 MaxPool
-3x3 32 units Convolutional kernel
- 4x4 MaxPool
- 64 units dense layer (ReLU)
- 29 units output layer (SoftMax)

Batch Size: 64
Optimizer: Adam
Learning Rate: 0.001

Accuracy Train: 0.90
Accuracy Val: 0.61

It improves the maximum learning corresponding to The validation set. Overfitting is not a problem for the moment.
However, the learning curves are quite noisy.





#4: Noisy learning curve: BN and lower LR

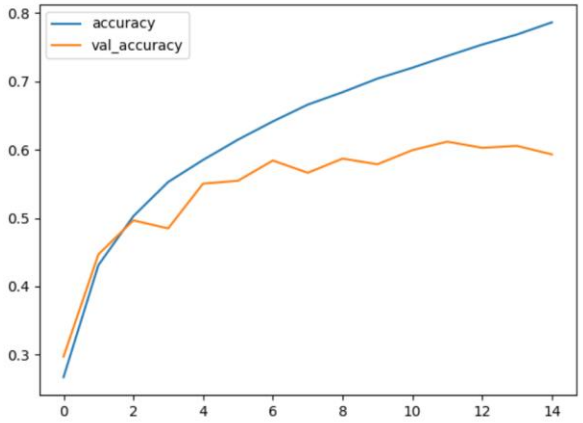
In #3 the learning curves where noisy. This Can make results less trustworthy or due to Luck.

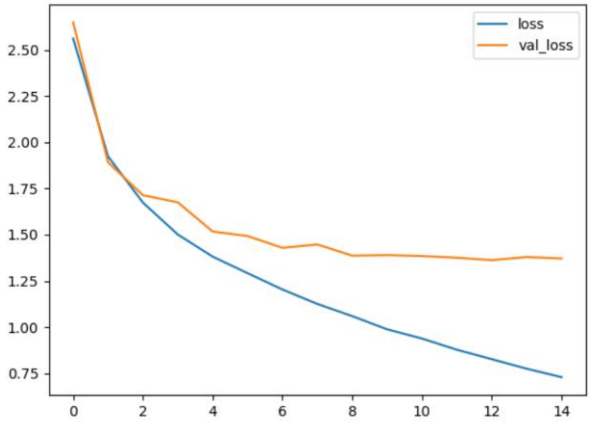
- 3x3 16 units Convolutional kernel (ReLU)
-4x4 MaxPool
-Batch Normalization
-3x3 32 units Convolutional kernel
- 4x4 MaxPool
- 64 units dense layer (ReLU)
- 29 units output layer (SoftMax)

Batch Size: 64
Optimizer: Adam
Learning Rate: 0.00005

Accuracy Train: 0.72
Accuracy Val: 0.58

In #4 learning rate is reduce to avoid this Phenomenon. However, while lowering down The learning rate, the model stopped learning Batch normalization

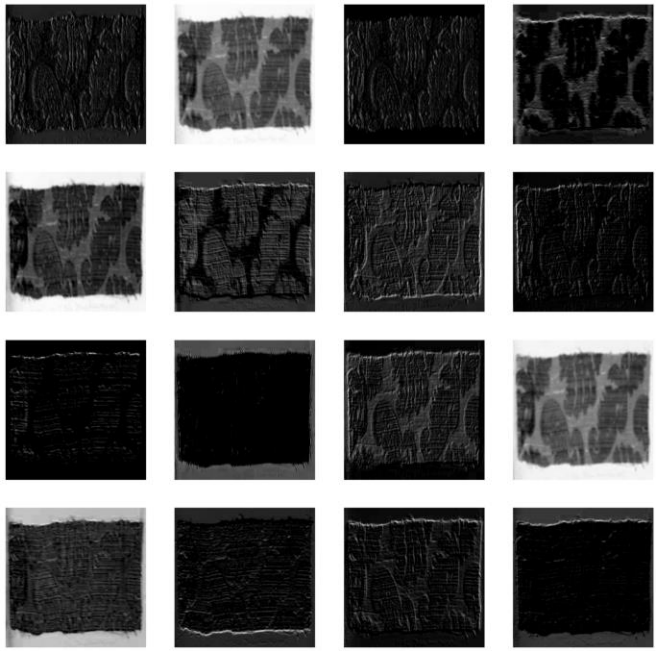




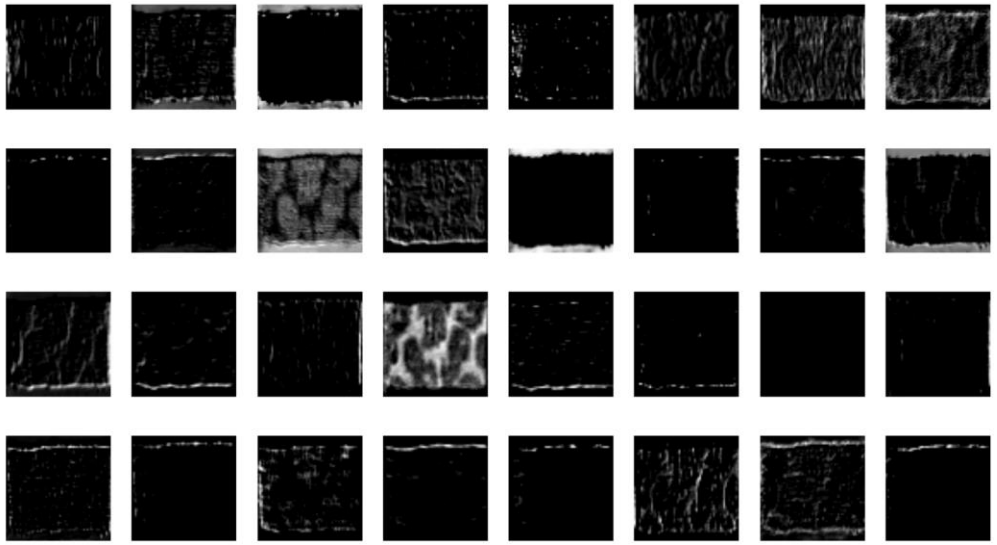
#4: Noisy learning curve: BN and lower LR

Input

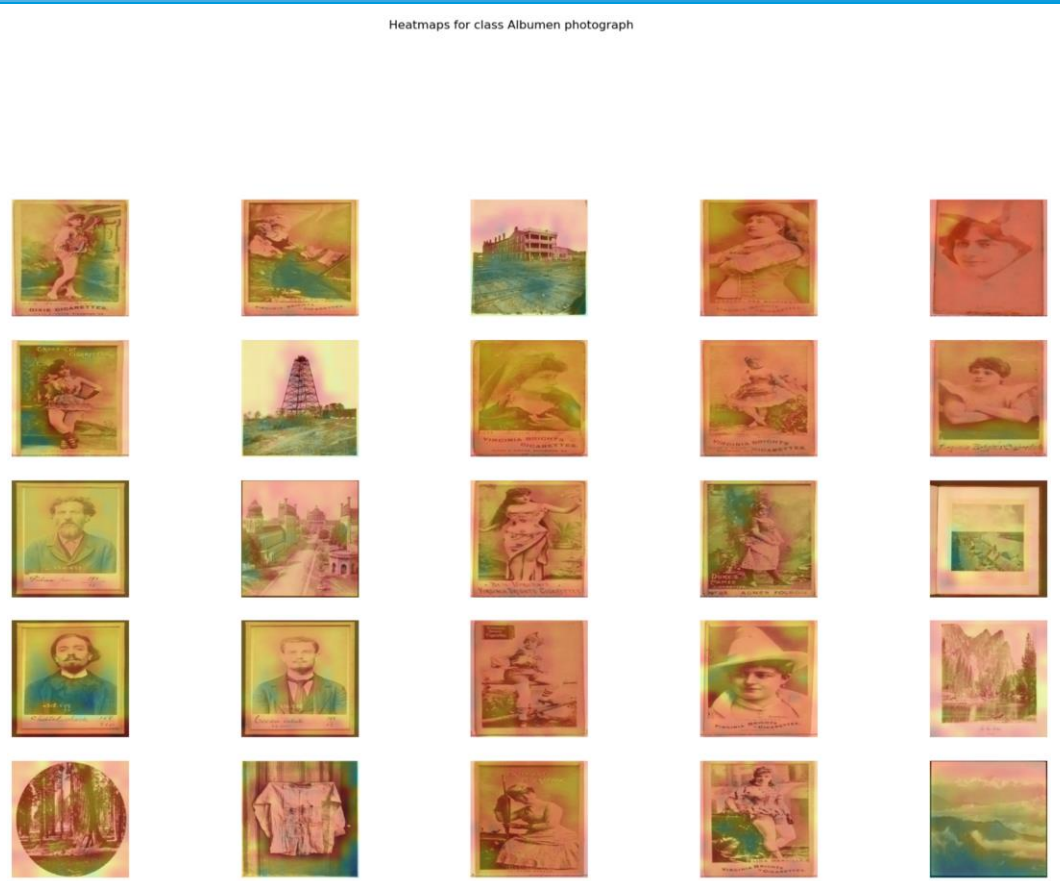
First Layer Activations



Second Layer Activations



Overfitted models focus on small details



#5: More Capacity: 3rd Convolutional Layer

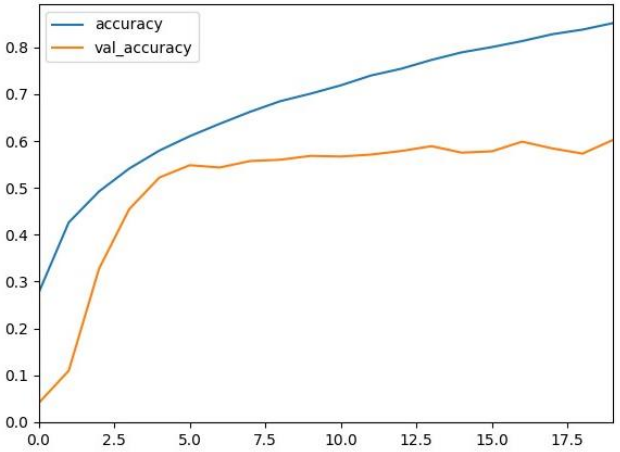
Added more progressive complexity and as wanted, we reach a strong overfitting

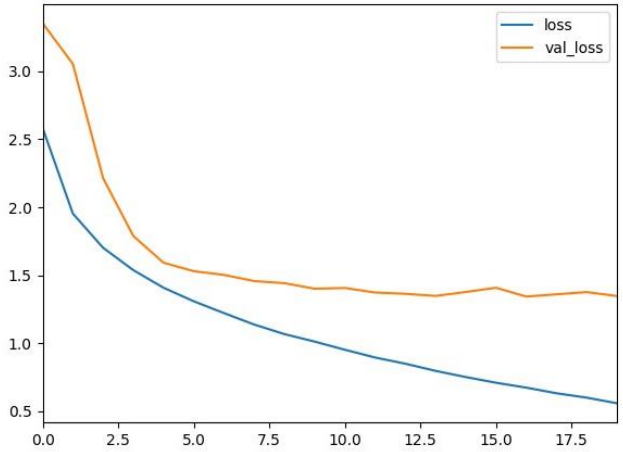
- 3x3 16 units Convolutional kernel (ReLU)
-4x4 MaxPool
-Batch Normalization
-3x3 32 units Convolutional kernel
- 2x2 MaxPool
-Batch Normalization
-3x3 64 units Convolutional kernel
-2x2 MaxPool
- 64 units dense layer (ReLU)
- 29 units output layer (SoftMax)

Batch Size: 128
Optimizer: Adam
Learning Rate: 0.00005

Accuracy Train: 0.86
Accuracy Val: 0.60

The smoother learning curve is mainly due to the
larger batch size.
The higher complexity of the model made possible a slight improvement in the accuracies





In many images we can see tha the attention of the model is the background

#6: Complete training saturation

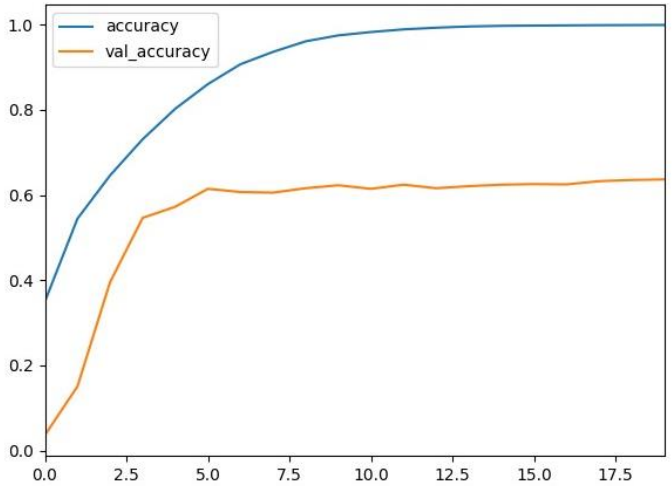
We reached quickly 100% training accuracy introducing 4th convolutional layer of 128 filters

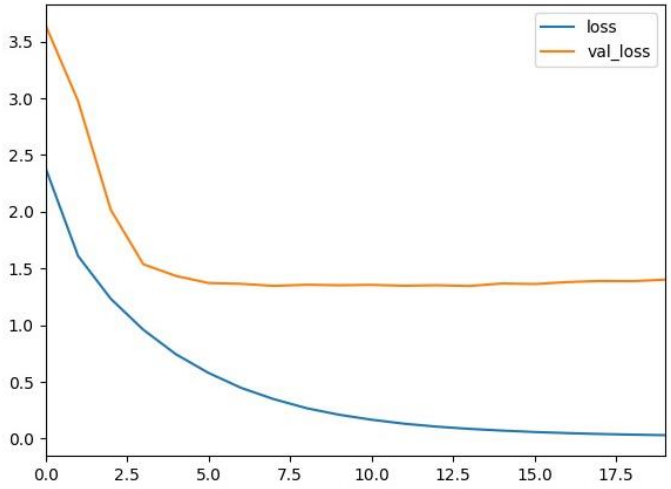
- 3x3 16 units Convolutional kernel (ReLU)
-2x2 MaxPool + BN
-3x3 32 units Convolutional kernel
- 2x2 MaxPool + BN
-3x3 64 units Convolutional kernel
-2x2 MaxPool + BN
-3x3 128 units Convolutional kernel
-2x2 MaxPool + BN
- 64 units dense layer (ReLU)
- 29 units output layer (SoftMax)

Batch Size: 128
Optimizer: Adam
Learning Rate: 0.00005

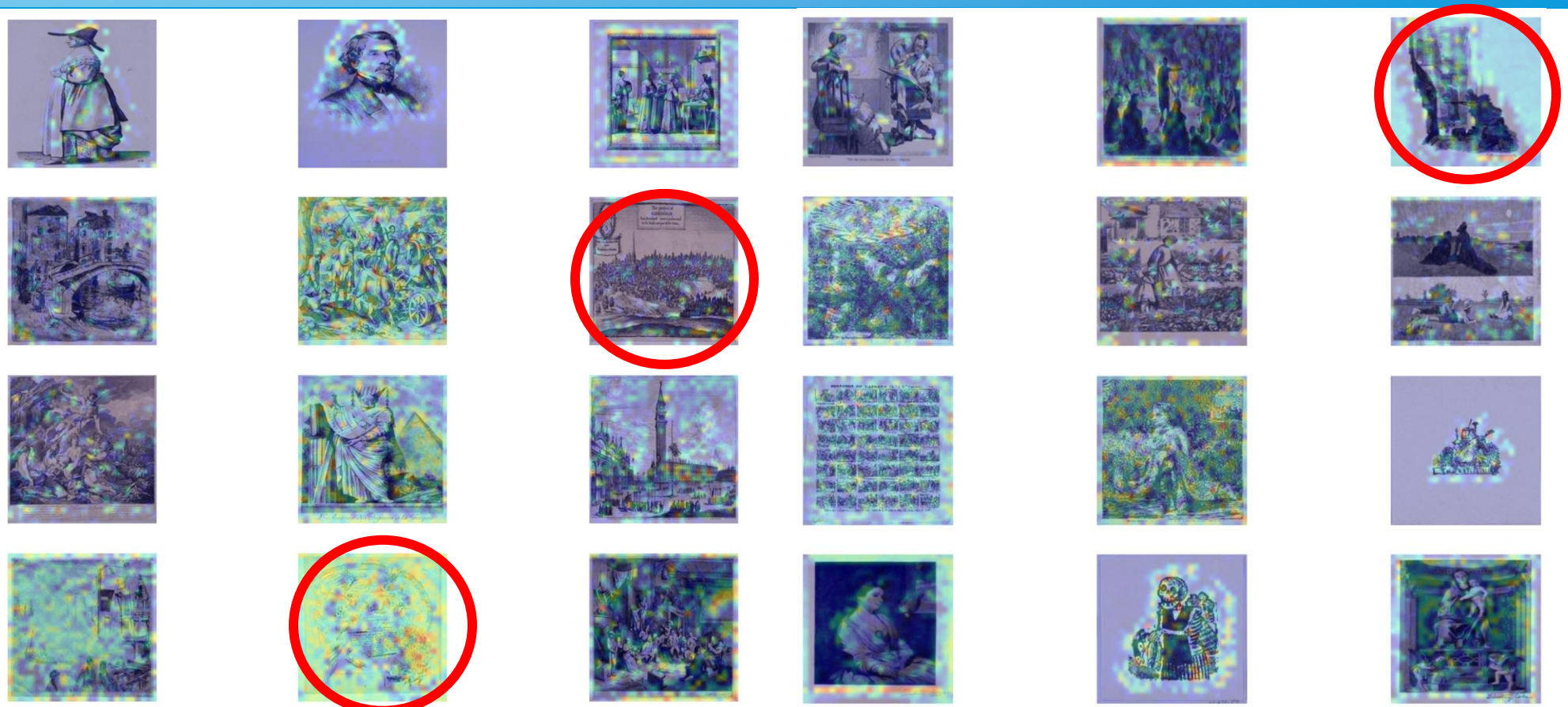
Accuracy Train: 0.99
Accuracy Val: 0.63

Our approach has been to reach strong overfitting before starting to regularize the model, thinking that once we start regularizing the accuracy for sure will not increase. In further experiments we plan to regularize and strike a balance between complexity and regularization.





For some samples the attention of the model is really low and background focused



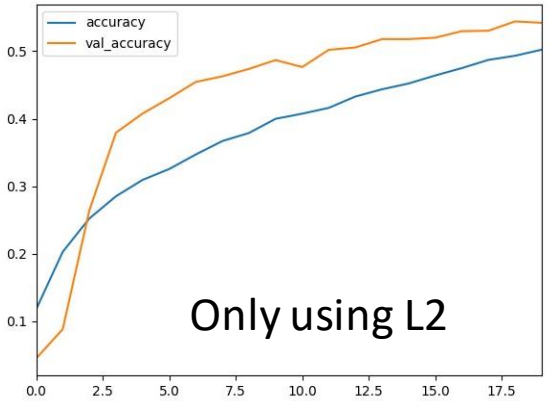
#7: First regularizations

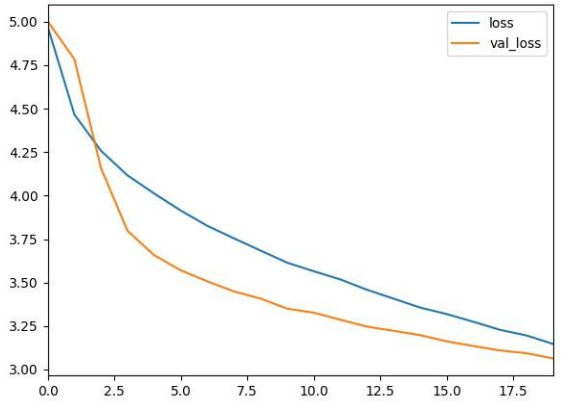
Starting to gradually add regularization to the model....

- 3x3 16 units Convolutional kernel (ReLU) + L2
-2x2 MaxPool + BN
-3x3 32 units Convolutional kernel + L2
- 2x2 MaxPool + BN
-3x3 64 units Convolutional kernel + L2
-2x2 MaxPool + BN
-3x3 128 units Convolutional kernel + L2
-2x2 MaxPool + BN
- 64 units dense layer (ReLU)
-DropOut
- 29 units output layer (SoftMax)

Batch Size: 128
Optimizer: Adam
Learning Rate: 0.00005

Accuracy Train: 0.85
Accuracy Val: 0.64

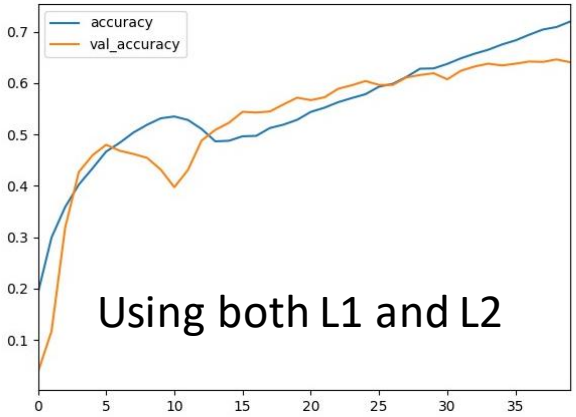


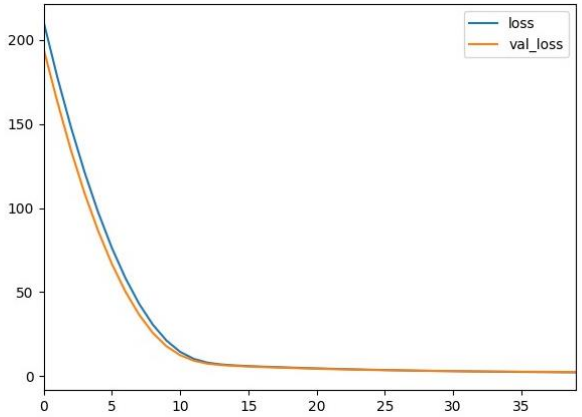


We introduced L1 and L2 regularization, less in the firsts layers and more in subsequent layers because that is where the complexity lies.
L1 had higher values because it has been put in larger layers in order to allow the model to do feature selections and reduce the important filters

W.r.t #6 the two curves are growing together and the training is taking more time because of the penalization introduced.

Continuation of #7





After introducing L1 in Deeper layers we noted that the accuracy had a sudden drop and recovery at the beginning of the training and this might be explained by some filters weights going to zero

We think more regularization is needed but that this is the path to follow.
It may be necessary to increase complexity if after introducing some more regularization if the training results too much hard for the model
In next experiments we are going to increment L1 and L2 values, as well as introducing Dropout layers after the Convolutional layers.

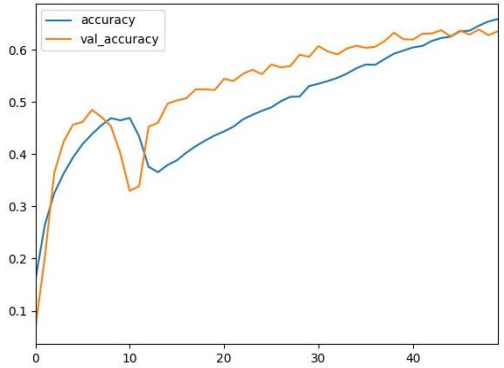
#8: More regularization

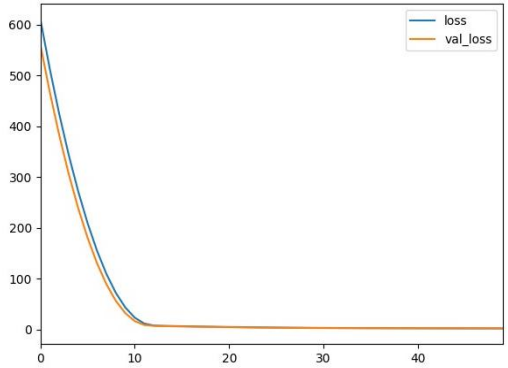
Training accuracy is more in line with validation accuracy but still overfitting: We continue to regularize

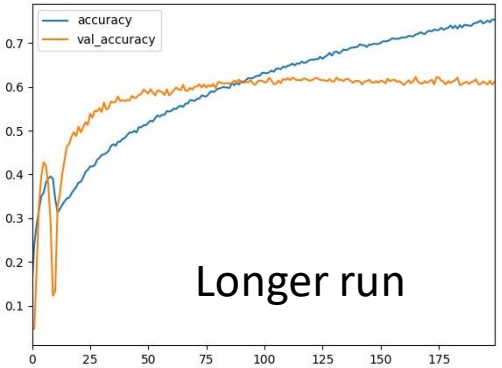
- 3x3 16 units Convolutional kernel (ReLU) + L2
-2x2 MaxPool + BN
-3x3 32 units Convolutional kernel + L2
- 2x2 MaxPool + BN
-3x3 64 units Convolutional kernel + L1 + L2
-2x2 MaxPool + BN
-3x3 128 units Convolutional kernel + L1 + L2
-2x2 MaxPool + BN
- 64 units dense layer (ReLU)
-DropOut
- 29 units output layer (SoftMax)

Batch Size: 128
Optimizer: Adam
Learning Rate: 0.00005

Accuracy Train: 0.77
Accuracy Val: 0.60







Longer run

Comments:
- Learining curve less steep L1 used for feature selection in bigger layers
- Increase regularization hyperparameters values

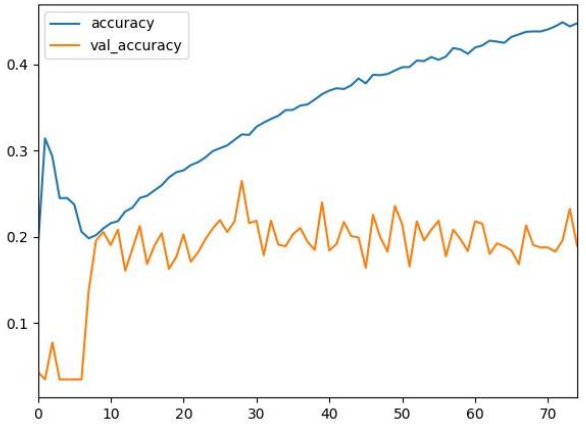
#9: Too much regularization

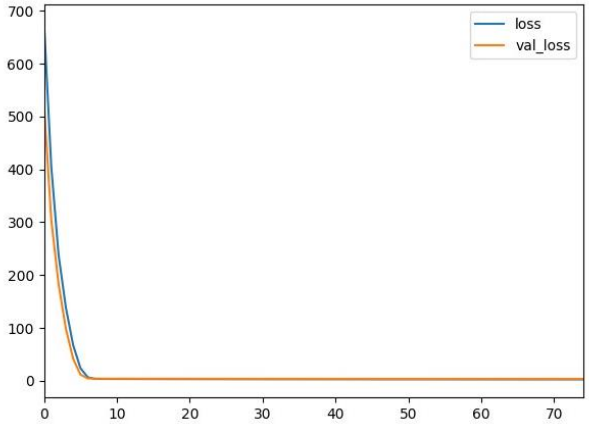
Added Dropout layers with low values at the beginning and higher values to the final layers

- 3x3 16 units Convolutional kernel (ReLU) + L2
-2x2 MaxPool + BN + Dropout
-3x3 32 units Convolutional kernel + L2
- 2x2 MaxPool + BN + Dropout
-3x3 64 units Convolutional kernel + L1 + L2
-2x2 MaxPool + BN + Dropout
-3x3 128 units Convolutional kernel + L1 + L2
-2x2 MaxPool + BN + Dropout
- 64 units dense layer (ReLU)
-DropOut
- 29 units output layer (SoftMax)

Batch Size: 128
Optimizer: Adam
Learning Rate: 0.00005

Accuracy Train: 0.45
Accuracy Val: 0.21





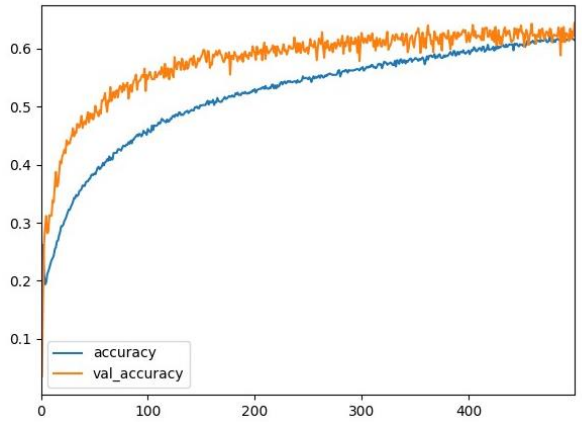
Combined effect of Dropout (0.3/0.6), L1 (0.1) and L2 (0.01/0.001) were too much and shifted the equilibrium towards a too simple model that is also very slow to learn due to eccessive penalizations.
Thus we add more capacity or reduce regularization and try to reach a better trade off

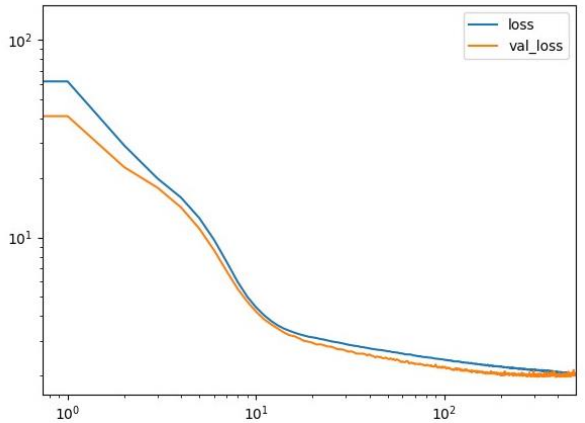
#10: Finding a balance

We slightly enlarged the capacity of the network doubling the filters of the first layer

- 3x3 32 units Convolutional kernel (ReLU) + L2
-2x2 MaxPool + BN
-3x3 32 units Convolutional kernel + L2
- 2x2 MaxPool + BN
-3x3 64 units Convolutional kernel + L1 + L2
-2x2 MaxPool + BN
-3x3 128 units Convolutional kernel + L1 + L2
-2x2 MaxPool + BN
- 64 units dense layer (ReLU) + L1 + L2
-DropOut
- 29 units output layer (SoftMax)

Batch Size: 128
Optimizer: Adam
Learning Rate: 0.00005





W.r.t #9 we doubled the number of filters in the firs layers and reduced L1 to 0.01, L2 to 0.001 and Dropout to 0.6
The result is a quite slow training due to the higher complexity of the model but a more common trend between the accuracies and the losses. The lossess are shown in log log scale to visualize them better. We kept the experiment running until the improvement was relevant.

Accuracy Train: 0.73
Accuracy Val: 0.63

Still improvable but many advancement can be noticed

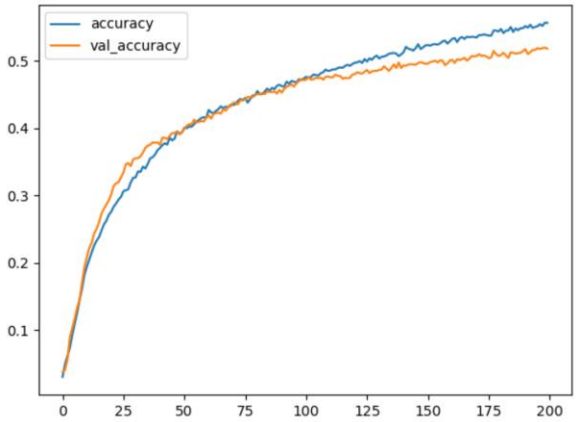
#11: Introducing Image Augmentation

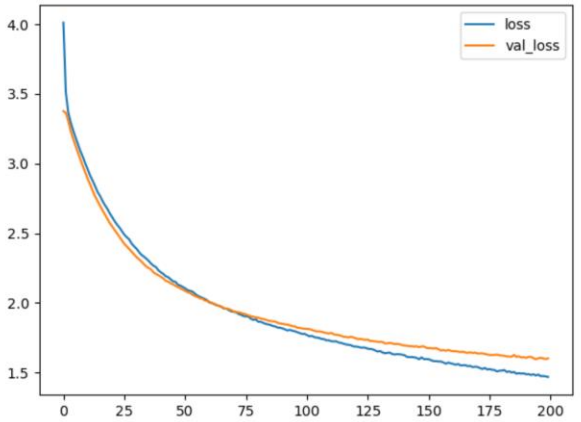
In previous #, even with low learning rates Overfitted came quickly because the model Identified too specific features in train and validation

- 3x3 32 units Convolutional kernel (ReLU)
- 4x4 MaxPool + Batch Normalization
- 3x3 64 units Convolutional kernel (ReLU)
- 4x4 MaxPool + Batch Normalization
- 3x3 128 units Convolutional kernel (ReLU)
- 4x4 MaxPool
- 512 units dense layer (ReLU)
- 64 units dense layer (ReLU)
- 29 units output layer (SoftMax)

Batch Size: 128
Optimizer: Adam
Learning Rate: 2*10^⁻6

Image augmentation includes random rotations (-30, 30) range, random shifts and random Horizontal flips.
Overfitting takes much longer to happen and similar resultsare obtained





Accuracy Train: 0.54
Accuracy Val: 0.52

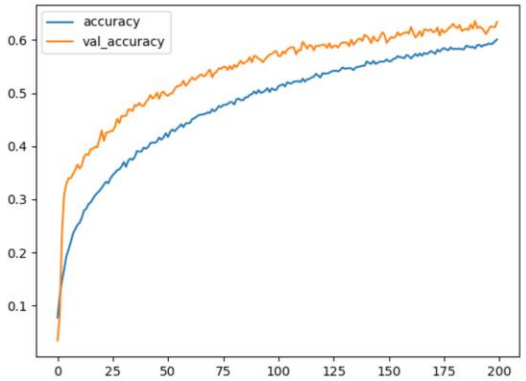
#12: Use architecture #7 with Augmentation

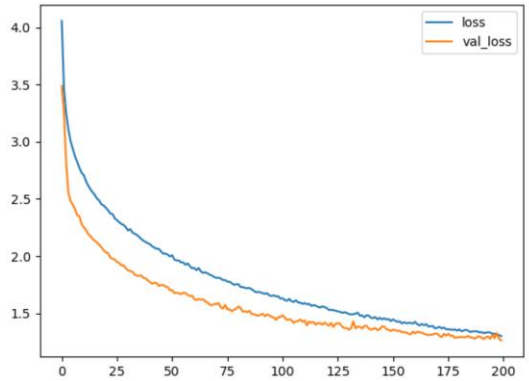
In #7 a deep convolutional architecture was Used together with regularization to avoid Overfitting. In this case, instead of introducing Regularization in conv layers, image Augmentation is performed

- 3x3 16 units Convolutional kernel (ReLU)
- 2x2 MaxPool + Batch Normalization
- 3x3 32 units Convolutional kernel (ReLU)
- 2x2 MaxPool + Batch Normalization
- 3x3 64 units Convolutional kernel (ReLU)
- 2x2 MaxPool
- 3x3 128 units Convolutional kernel (ReLU)
- 2x2 MaxPool
- 512 units dense layer (ReLU)
- 64 units dense layer (ReLU)
- 29 units output layer (SoftMax)

Batch Size: 128
Optimizer: Adam
Learning Rate: 1*10^⁻5

Slow training with data Augmentation and Deep Feature Extractors and Classifier allowed The model to learn only common patterns in train And validation.



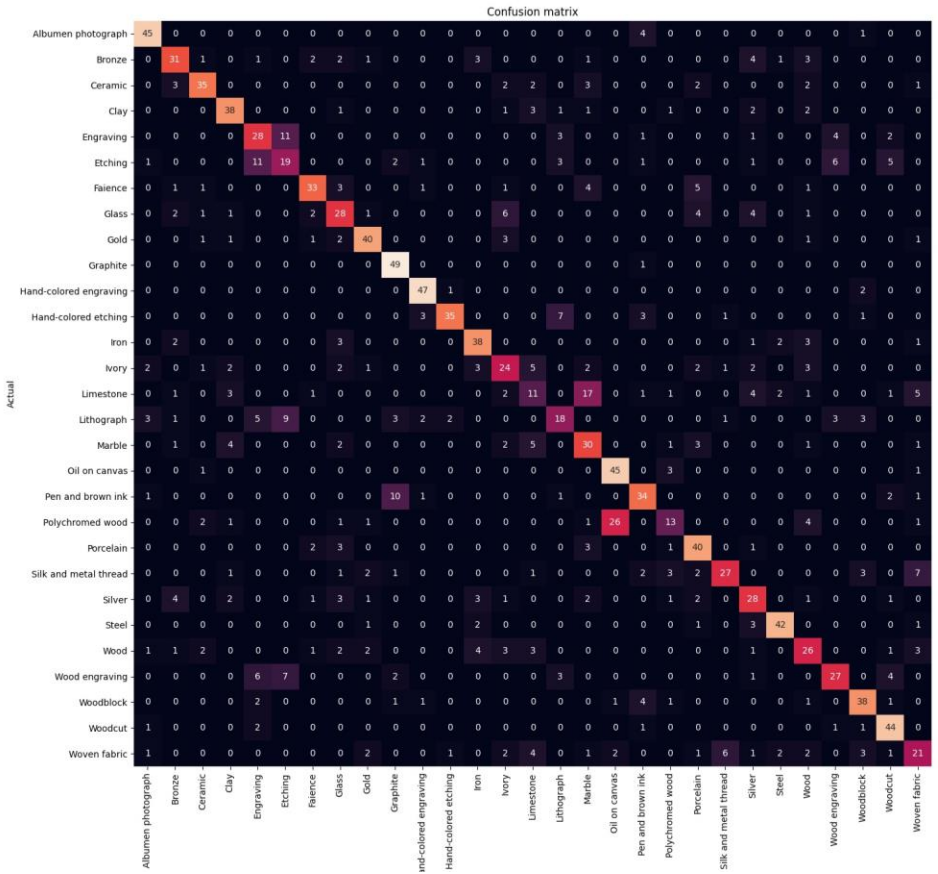


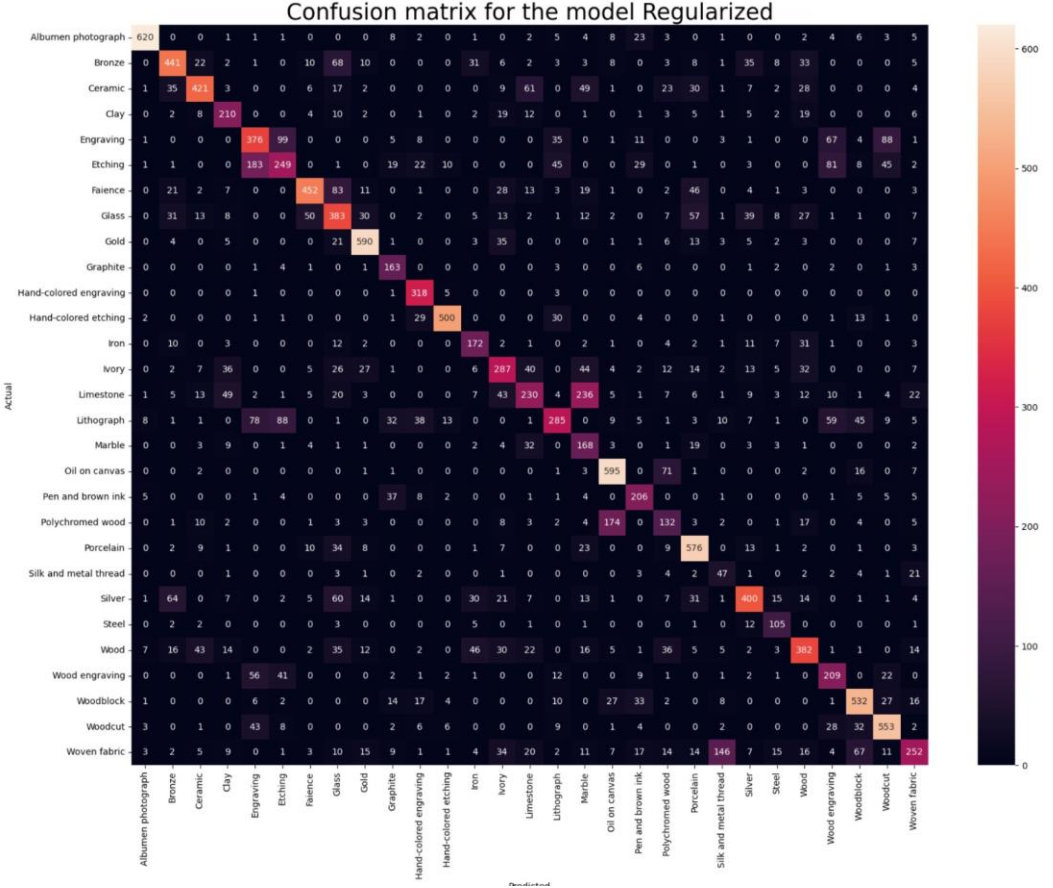
Accuracy Train: 0.59
Accuracy Val: 0.63

#: Test Results

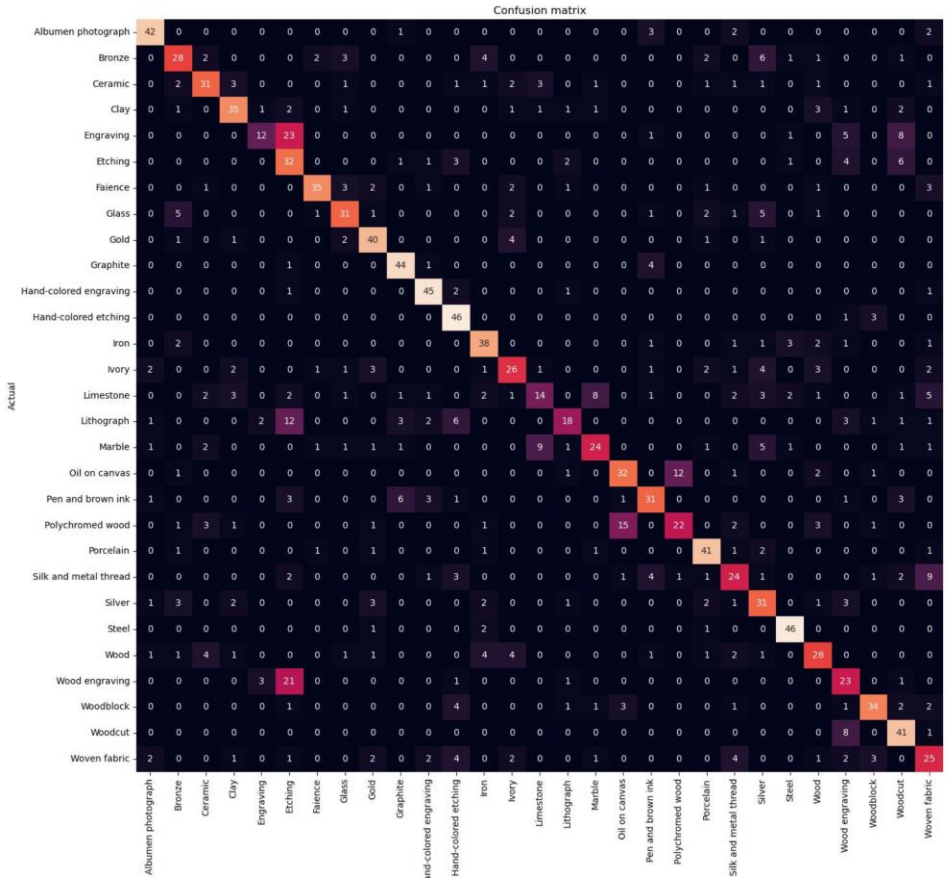
Model #	Train	Validation	Test	Overfitting
1	0.97	0.307	0.27	High
2	0.66	0.45	0.43	High
3-4	0.72	0.58	0.58	Medium
5-6	0.86	0.62	0.53	Medium
8	0.77	0.60	0.64	Medium
10	0.73	0.64	0.63	Medium
11	0.54	0.52	0.51	Low
12	0.59	0.63	0.62	Low
13	0.75	0.67	0.67	Medium

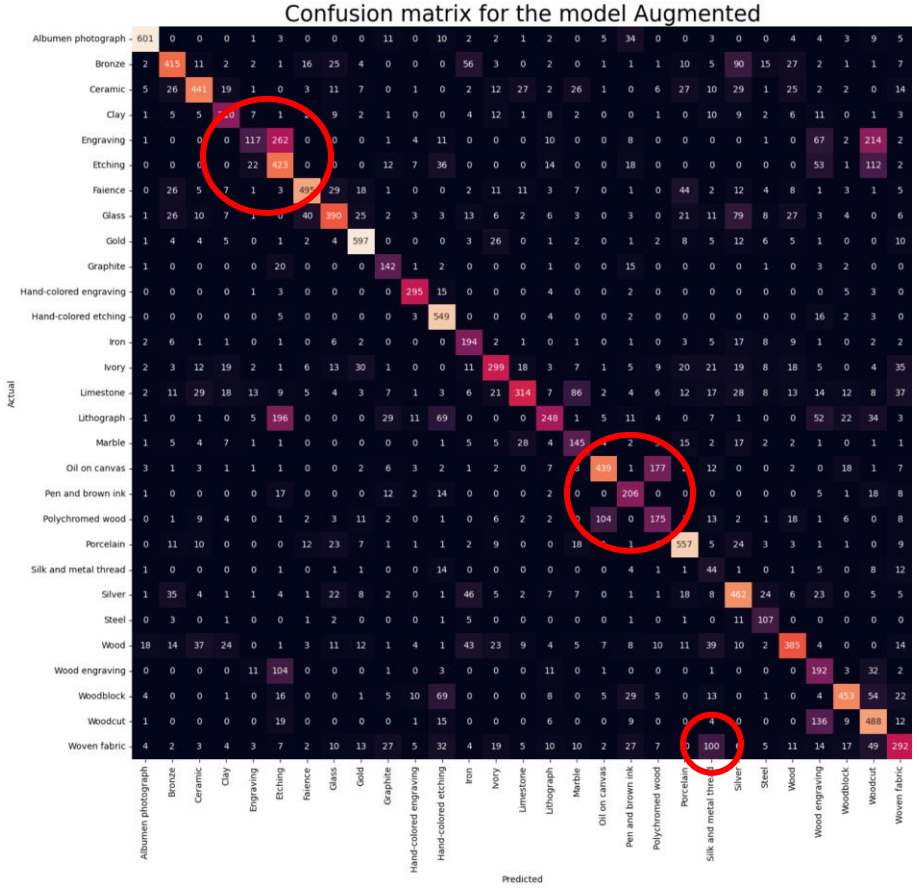
Test Results Model #10 – Regularized





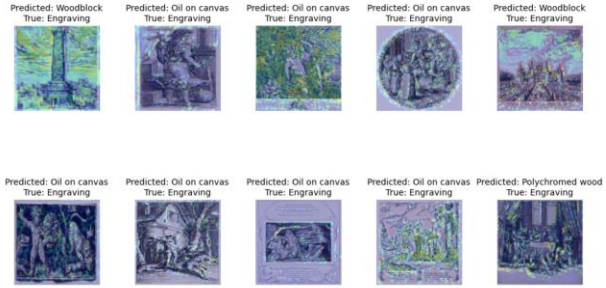
Test Results Model #12 - Augmented





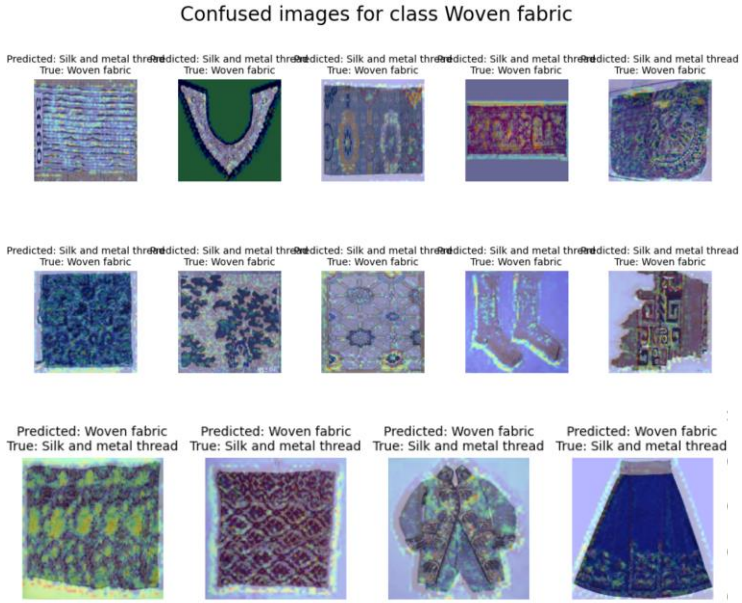
Most confused classes Test

Engraving

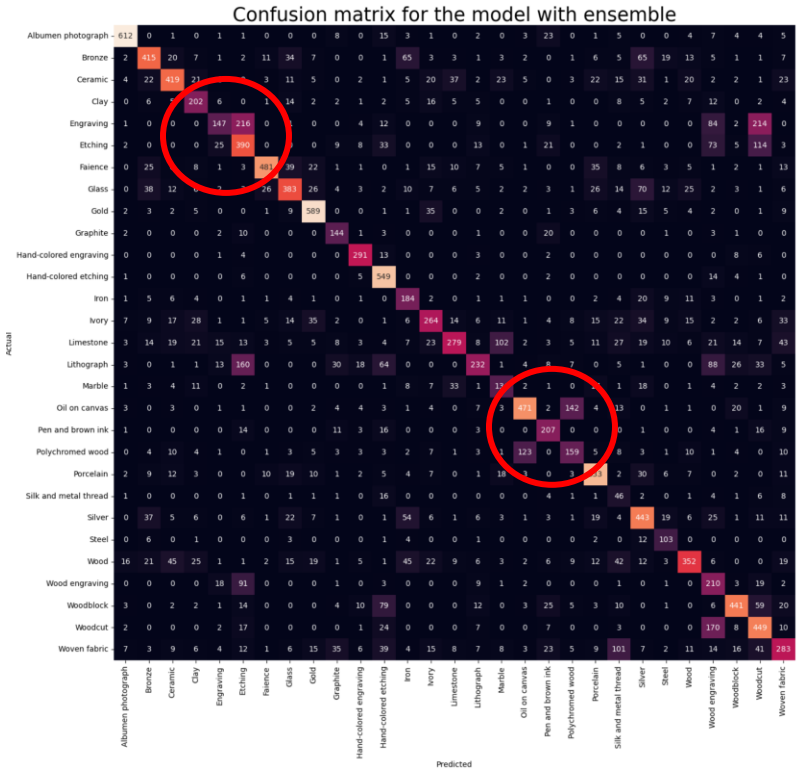


Etching

Woven Fabric and Silk and Metal Thread

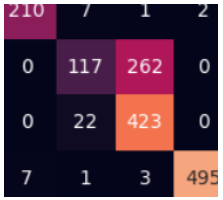


Result of Ensemble

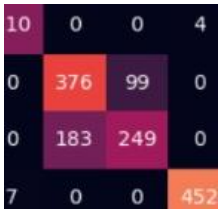


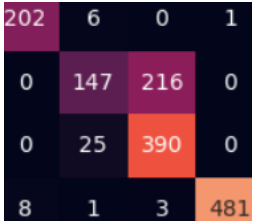
Two interesting facts of the ensemble that are the cause of the improved final accuracy

Augmented



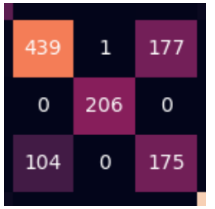
Regularized



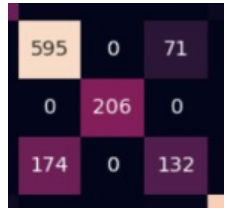


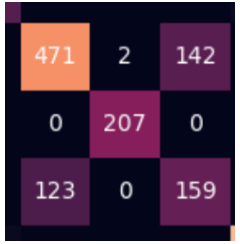
Ensemble

Augmented



Regularized





Ensemble

Further Improvements

- Use of pretrained models for feature extraction
- Using width and height information from metadata
- Use Ridge regression to calculate weights for each class
- Train a general model for parent categories and specialist models for subcategories

References

[1] Parés, F., Arias-Duart, A., Garcia-Gasulla, D. et al. The MAMe dataset: on the relevance of high resolution and variable shape image properties. Appl Intell 52, 11703–11724 (2022). https://doi.org/10.1007/s10489-021-02951-w

[2] Shorten, C., Khoshgoftaar, T.M. A survey on Image Data Augmentation for Deep Learning. J Big Data 6, 60 (2019). https://doi.org/10.1186/s40537-019-0197-0

[3] Chollet, F. Grad-CAM class activation visualization (2021) https://keras.io/examples/vision/grad_cam/

[4] Shima, Y. Image Augmentation for Object Image Classification Based On Combination of Pre-Trained CNN and SVM. Journal of Physics: Conference Series. https://dx.doi.org/10.1088/1742-6596/1004/1/012001