Artificial Neural Networks Project

Arno Deceuninck

June 17, 2023

1 Settings used

The settings used for each model are shown in table 1. Different learning rates and number of epochs were tested. The final values performed well on all models, which is why they often have the same parameter values (except for the perturbation detection task, which has a lot less epochs). The feature part must remain the same for all modelse (as mentioned in the assignment), and that is why only the number of fully connected layers in the classifier part is shown.

Model	Input image size	Learning rate	Optimizer	Batch size	Number of epochs	Fully connected layers in classifier part
Scene classification (fully-supervised)	(224, 224, 3)	0.00001	Adam	15	15	1
Rotation classification	(224, 224, 3)	0.00001	Adam	15	15	1
Scene classification with rotation pretext	(224, 224, 3)	0.00001	Adam	15	15	1
Perturbation classification	(224, 224, 3)	0.00001	Adam	15	3	1
Scene classification with perturbation pretext	(224, 224, 3)	0.00001	Adam	15	15	1

Table 1: Settings used for the different models

2 Performance

The multi-label classification accuracy is given in table 2.

The multi-label confusion matrix of the supervised learning model is given in table **??**, for the classification after the rotation pretext task is given in table 4 and for the classification after the perturbation pretask in table 5.

Model	Accuracy
Scene classification (fully-supervised)	94%
Scene classification with rotation pretext	71%
Scene classification with perturbation pretext	70%

Table 2: Multi-scene classification accuracy

	bedroom	coast	forest	highway	industrial	insidecity	kitchen	livingroom	mountain	office	opencountry	store	street	suburb	tallbuilding
bedroom	102	0	0	0	0	0	1	12	0	1	0	0	0	0	0
coast	0	251	1	0	0	0	0	0	1	0	7	0	0	0	0
forest	0	0	220	0	0	0	0	0	4	0	4	0	0	0	0
highway	0	1	0	154	2	0	0	0	0	0	0	0	2	0	1
industrial	0	0	0	3	189	6	2	1	0	0	2	2	0	2	4
insidecity	0	0	0	2	4	193	1	0	0	0	0	1	5	1	1
kitchen	3	0	0	0	0	0	105	1	0	0	0	1	0	0	0
livingroom	11	0	0	0	0	1	3	170	0	4	0	0	0	0	0
mountain	0	1	5	0	0	0	0	0	261	0	7	0	0	0	0
office	0	0	0	0	0	0	1	2	0	112	0	0	0	0	0
opencountry	0	27	6	2	0	0	0	0	6	0	269	0	0	0	0
store	2	0	1	0	6	0	0	0	0	0	0	206	0	0	0
street	0	0	1	4	0	7	0	0	0	0	0	0	178	0	2
suburb	0	0	0	0	0	0	0	0	0	0	0	0	0	141	0
tallbuilding	0	0	1	0	3	5	0	0	1	0	0	0	2	0	244

Table 3: Confusion matrtix for multi-label scene classification with the supervised learning model

	bedroom	coast	forest	highway	industrial	insidecity	kitchen	livingroom	mountain	office	opencountry	store	street	suburb	tallbuilding
bedroom	87	0	1	0	1	0	3	14	0	6	2	0	0	2	0
coast	0	192	3	16	1	0	0	0	11	0	36	0	0	1	0
forest	0	3	169	0	1	0	0	1	18	0	27	3	1	5	0
highway	0	9	2	128	2	1	0	0	3	1	3	3	4	2	2
industrial	1	0	1	14	84	11	10	6	2	2	13	19	15	6	27
insidecity	2	0	2	4	3	131	13	1	0	3	1	12	21	8	7
kitchen	4	0	1	2	1	1	71	9	0	12	0	3	2	4	0
livingroom	38	0	0	0	0	0	23	94	1	19	0	8	2	4	0
mountain	0	12	8	1	0	0	1	0	203	0	48	0	0	0	1
office	2	0	0	4	0	1	16	7	0	84	0	0	0	0	1
opencountry	0	37	12	10	0	0	1	1	23	0	220	0	0	5	1
store	1	0	5	0	0	4	18	2	1	6	0	160	14	2	2
street	0	0	0	4	2	10	1	0	0	1	0	8	163	0	3
suburb	0	0	1	0	0	1	1	0	0	0	0	1	0	135	2
tallbuilding	0	1	6	0	6	8	7	1	1	2	3	8	2	1	210

Table 4: Confusion matrtix for multi-label scene classification with the self-supervised learning model with the rotation pretext

	bedroom	coast	forest	highway	industrial	insidecity	kitchen	livingroom	mountain	office	opencountry	store	street	suburb	tallbuilding
bedroom	92	0	0	0	1	0	5	11	1	3	0	0	0	2	1
coast	1	150	5	17	2	1	2	0	18	1	57	0	0	1	5
forest	0	10	159	1	3	1	0	0	22	0	26	1	1	3	1
highway	0	2	1	126	3	2	1	0	2	2	7	2	4	1	7
industrial	0	3	1	7	115	13	8	0	1	3	7	10	19	7	17
insidecity	0	2	1	6	8	130	5	1	4	4	1	4	17	11	14
kitchen	11	0	0	0	1	0	79	7	0	3	0	5	2	2	0
livingroom	41	0	0	0	1	0	7	105	1	18	0	4	7	5	0
mountain	0	14	10	7	1	1	0	2	204	0	33	0	0	0	2
office	6	0	0	0	0	0	1	4	0	104	0	0	0	0	0
opencountry	0	41	35	14	0	0	0	0	43	0	171	1	1	1	3
store	2	1	1	0	0	2	17	3	1	8	0	171	7	0	2
street	1	0	0	6	5	19	1	1	2	2	0	2	146	0	7
suburb	0	0	0	0	0	4	1	1	0	0	0	0	1	130	4
tallbuilding	1	1	4	7	10	10	0	0	3	1	2	1	7	6	203

Table 5: Confusion matrtix for multi-label scene classification with the self-supervised learning model with the perburtation pretext

3 Performance comparison

The supervised model had the highest classification accuracy (94%). This is probably because this model could train a lot more features and the features from efficientnet after the pretext task were not useful enough to freeze them and only rely on the classification layer.

The accuracy of the scene classification after the rotation pretext task (71%) was slightly better than the accuracy after the perturbation pretext task (70%), probably because determining the right rotation requires a model to look more at actual features in the model instead. The perturbation pretext task does not really require a model to look at the features. However, a larger difference between the two would be expected because of this, which was not the case.

4 Overfitting strategies

To prevent underfitting and overfitting, the loss on both the train and validation set was monitored after each epoch. As long as the validation loss keeps decreasing, we continue with the next epoch, since the model is still underfit. If only the train loss is decreasing, but the validation loss is increasing for multiple consecutive epochs in a row, the model is overfitting and the model before the validation loss started increasing.

Different learning rates were also evaluated, since setting the learning rate too low increases the likelihood of underfitting, since this would need a lot more epochs. Setting this learning rate too high might prevent reaching an optimal model.

5 Score-CAM explanations

The explanations for the supervised model are shown in figure 1, for the model after the rotation pretext in figure 2 and for the model after the perturbation pretext in figure 3. In all models, the first block contains low-level, high-resolution images. This block often contains highlights at the same place of the original image, but sometimes those highlights are inverted or some of those images contain more contrast than the original image. The second block also looks similar for all model, being the

original images with more focus on lines and contrast. The last convolutional layer focuses more on the higher-level features that were used by the models and differ more between the different models.

The self-supervised model mainly highlights the actuual object, similar as humans would highlight is. The model after the rotation pretext task contains more focsed highlights, with often less bright cells. The model after the perburbation pretext task looks more at objects more spread over the image (often the borders of the image). This is probably because the perbutations could block the actual part of interest when only looking at one specific location.

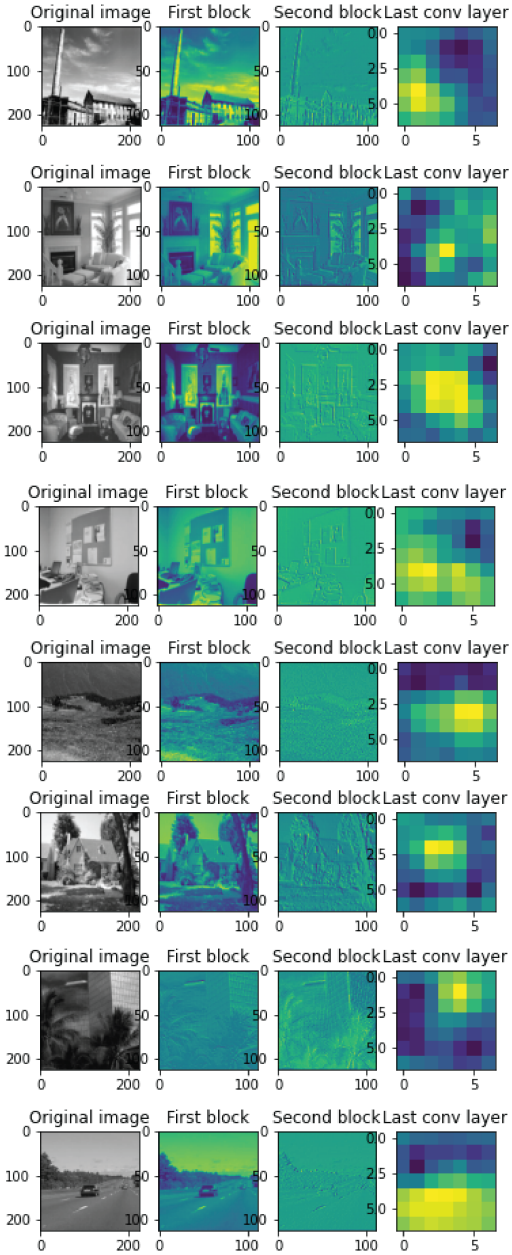

Figure 1: Score-CAM interpretation of correctly predicted images: Supervised model

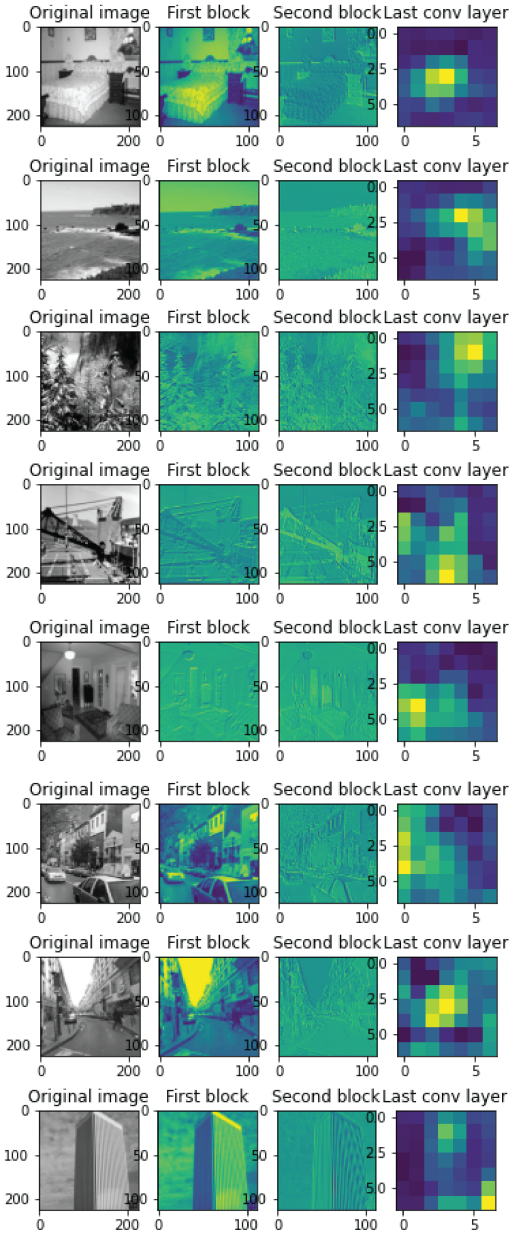

Figure 2: Score-CAM interpretation of correctly predicted images: Self-supervised model with rotation pretext task

6 Interpretation

The implementation of the interpretation task worked on the EfficientNet model itself, but not once the weights of my own fine-tuned models where loaded into it, which stopped the loss from decreasing and kept the image random pixels, as shown in image . The gradients of the random inputted did

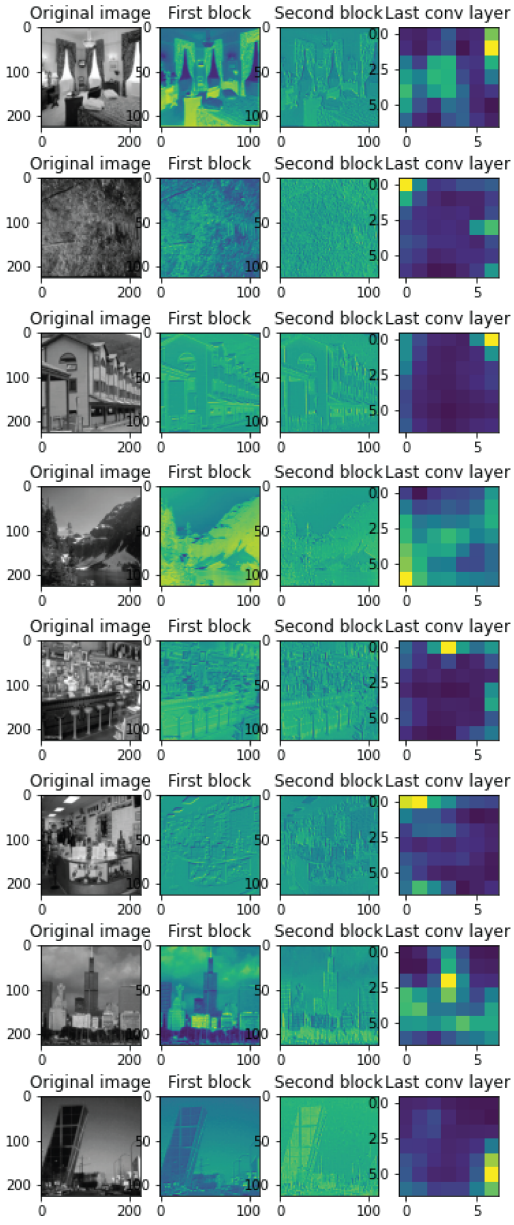

Figure 3: Score-CAM interpretation of correctly predicted images: Self-supervised model with rotation pretext task

not get calculated, keeping the image random pixels. When the model was put into training mode, the image changed a, bit more than random pixels, as shown in figure 6, but the model should not be in training mode (this was only for testing purposes). Requiring gradients for all layers also did not change anything.

The interpretation of the EfficientNet model itself actually contains recognizable parts, such as the interprtation of the "hen" (chicken) class in figure 4.

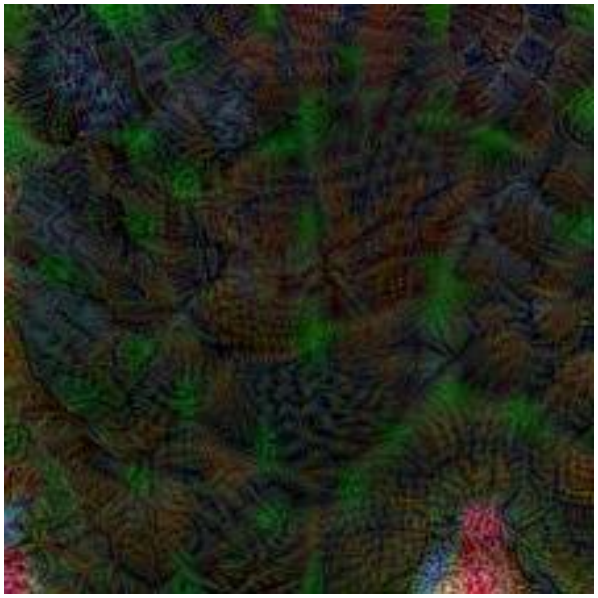

Figure 4: Interpretation of the "Hen" label in the EfficientNet model

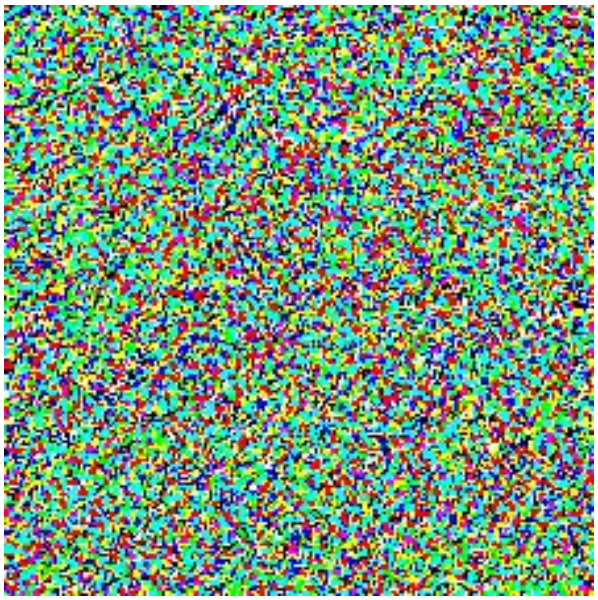

Figure 5: Failed interpretations of fine-tuned models

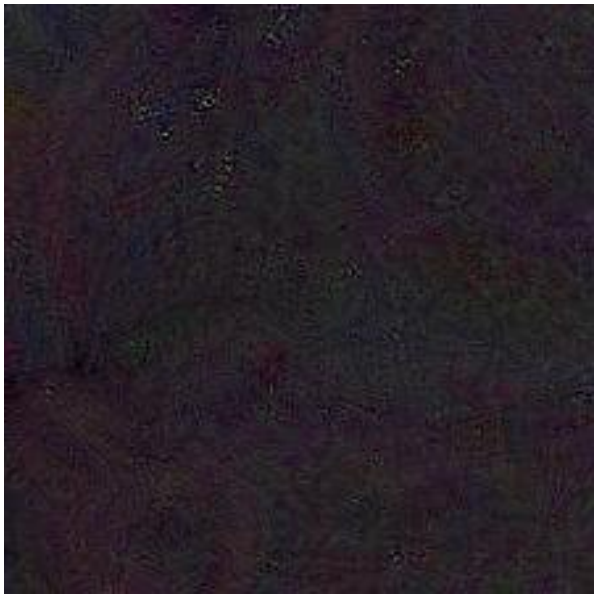

Figure 6: Enter Caption