Image classification with CNNs

David Štych
Aleksandra Jamróz

October 25, 2022

Baseline performance

Firstly, let us define a baseline to beat. We run the provided code with a 500 dataset size for 25 epochs. Maximum AUC was **0.804709** after 10 epochs. Then the model stopped improving noticeably.

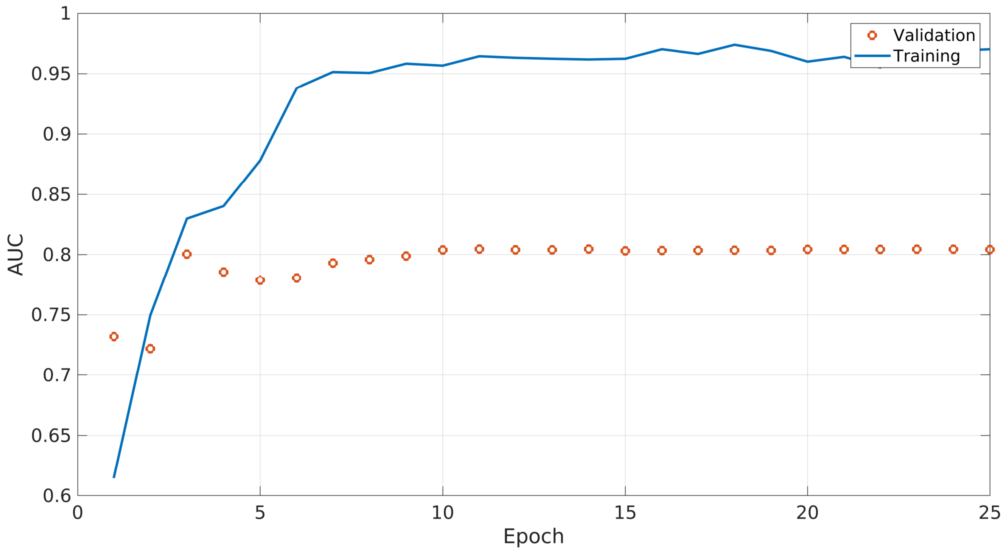

Figure 1: Default model - average AUC progress during learning

Improvements

- We have used data augmentation to randomize training data to reduce overfitting and achieve better performance [1]

  To the training dataset, we applied the following transformations:

  - Random rotation of the image by $\pm 30°$
    torchvision: RandomRotation [2]
  - Crop a random portion of an image and resize it to a given size
    torchvision: RandomResizedCrop [3]
  - Randomly flipping the image horizontally with a probability of 50%
    torchvision: RandomHorizontalFlip [4]
  - Converting to PyTorch tensor and normalization

To the validation and test dataset, we applied the following transformations:

  - Resize the image to a specific size
    torchvision: Resize [5]
  - Crop the image at the center to a specific size
    torchvision: CenterCrop [6]
  - Converting to PyTorch tensor and normalization

- Class balancing

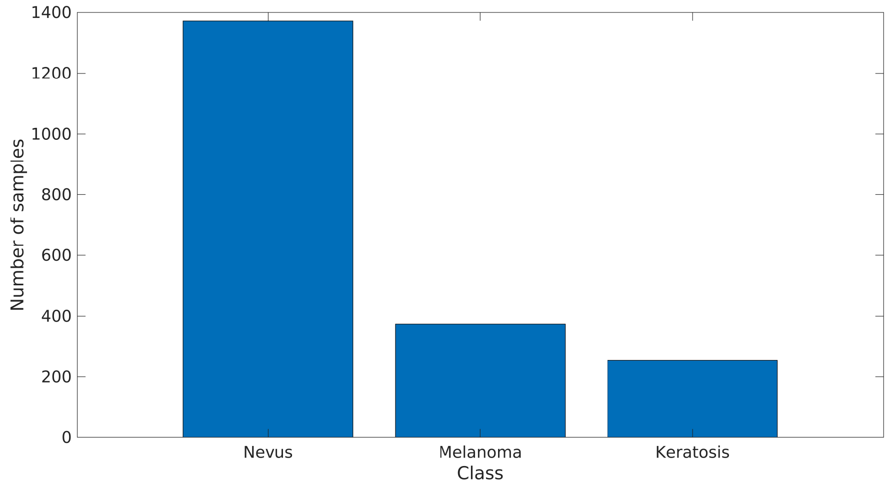

Figure 2: Dataset class distribution

The figure above shows the sample distribution in the dataset. Clearly, nevus is much more prevalent in the dataset, which is what we expect as it is a more common medical problem. [7] Nevertheless, we used class weights as an argument in the loss function, leading to an improvement of the model. For each class, we calculated weight with the following formula:

$$\text{Class weight} = 1 - \frac{\text{Number of samples in the dataset}}{\text{Total number of samples}}$$

- We have used Adam optimizer resulting in faster learning. Selected learning rate = 0.0001.

- We changed *maxSize* argument to 0, meaning we have used the entire dataset. That was possible thanks to the Czech Technical University in Prague (the home university of one of the authors of this report), which allowed us to use their computational resources. [8]

Conslusion

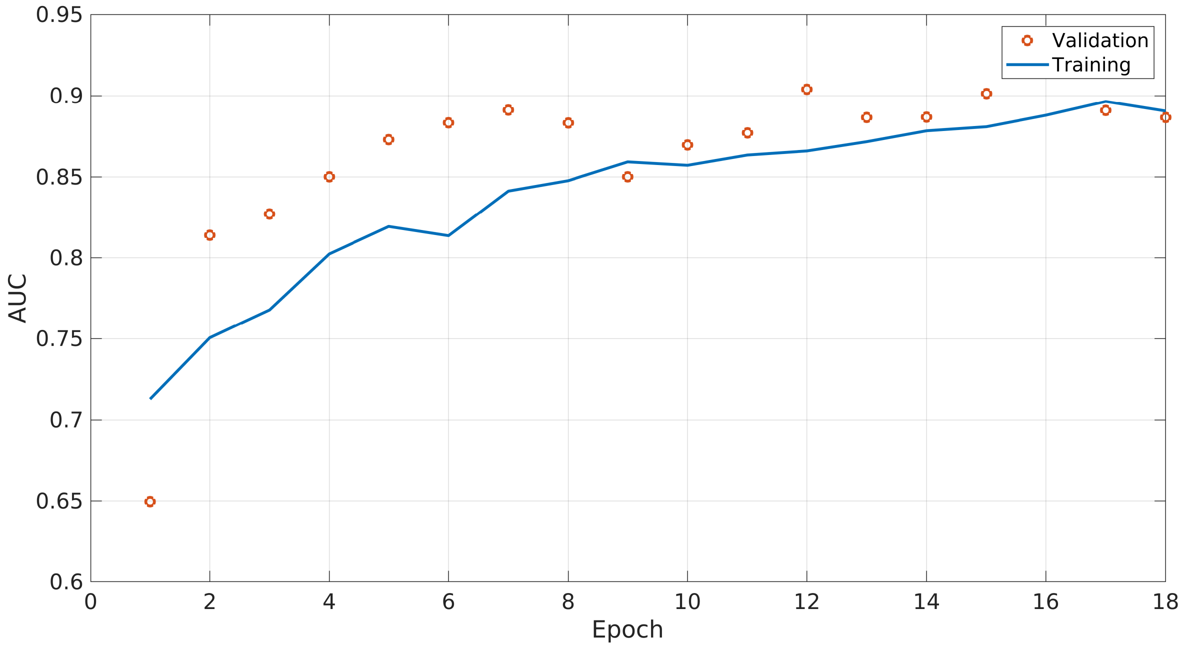

Figure 3: Final model - average AUC progress during learning

The figure above shows the development of AUC during learning. We tried to train to model for more epochs, but no further improvement was noticed after 15-16 epochs.

Our best result was **0.913973** average AUC over the validation dataset.

Changes to the model	**AUC - binary problem melanoma**	**AUC - binary problem keratosis**	**AUC Average**
No changes - baseline	0.731111	0.878307	0.804709
Data augumentation Using the entire dataset Adam optimizer	0.836389	0.954806	0.895597
Data augumentation Using the entire dataset Class balancing Adam optimizer	0.876667	0.951279	**0.913973**

References

[1] Ngoc Pham. Skin cancer classification with transfer learning in Pytorch. https://ngoc-pham.medium.com/skin-cancer-classification-with-transfer-learning-in-pytorch-389254e39ac1. Accessed: 2022-10-25.

[2] RandomRotation - Torchvision 0.13 documentation. https://pytorch.org/vision/stable/generated/torchvision.transforms.RandomRotation.html. Accessed: 2022-10-25.

[3] RandomResizedCrop - Torchvision 0.13 documentation. https://pytorch.org/vision/main/generated/torchvision.transforms.RandomResizedCrop.html. Accessed: 2022-10-25.

[4] RandomHorizontalFlip - Torchvision 0.13 documentation. https://pytorch.org/vision/stable/generated/torchvision.transforms.RandomHorizontalFlip.html. Accessed: 2022-10-25.

[5] Resize - Torchvision 0.13 documentation. https://pytorch.org/vision/stable/generated/torchvision.transforms.Resize.html. Accessed: 2022-10-25.

[6] CenterCrop - Torchvision 0.13 documentation. https://pytorch.org/vision/stable/generated/torchvision.transforms.CenterCrop.html. Accessed: 2022-10-25.

[7] M.D. Mikael Häggström. Incidence and malignancy of pigmented skin lesions. https://commons.wikimedia.org/wiki/File:Pie_chart_of_incidence_and_malignancy_of_pigmented_skin_lesions.png. Accessed: 2022-10-25.

[8] FEE CTU Prague - GPU servers. https://cyber.felk.cvut.cz/cs/study/gpu-servers/. Accessed: 2022-10-25.