Classification of Histopathology Images

Adam Aharony

Deep Learning, 2020 Spring Semester with Dr. Jonathan Rubin, Holon Institute of Technology.

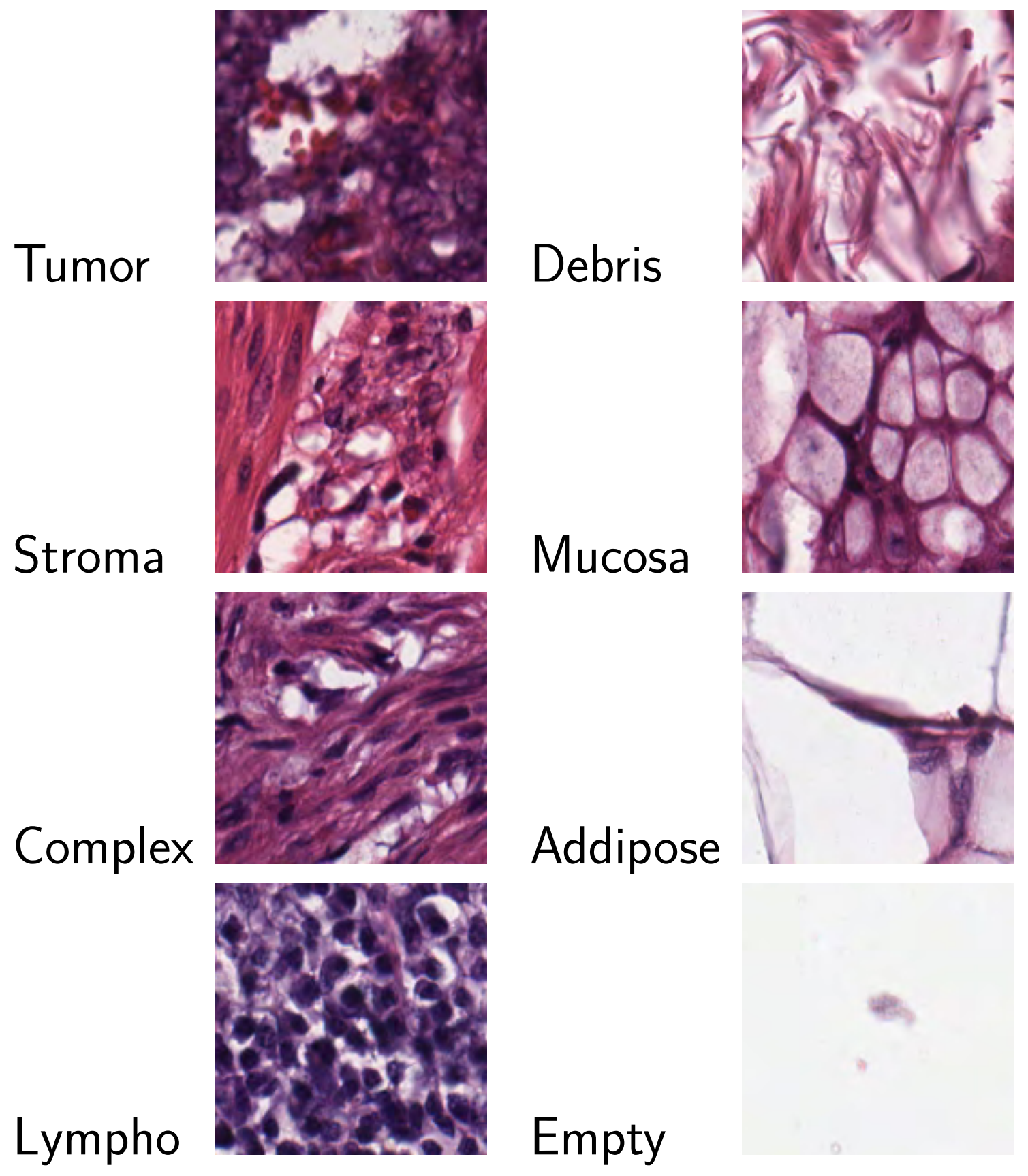
1. Abstract

As a part of the Deep Learning course in the spring semester of 2020 end-of-semester project, I was given the task of developing a model that is given an image of a biopsy scan, 5.000×5.000 pixels (RGB) in size.

The model needs to classify the type of cell(s) scanned within 8 categories (tumor, stroma, complex, lympho, debris, mucosa, addipose, empty), with a high-enough accuracy rate.

2. Background

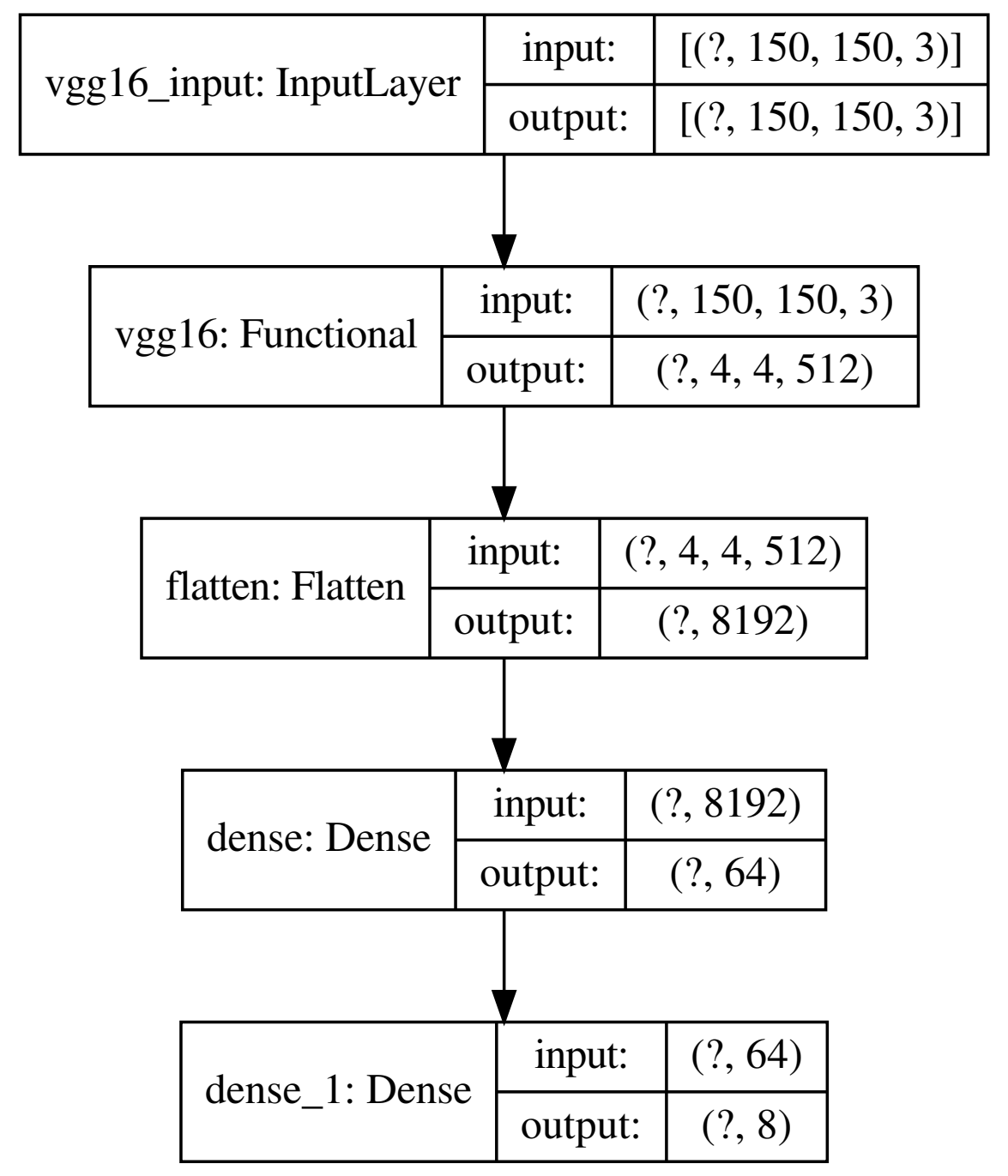
The train dataset is called *colorectal_histology* and available online in the Tensorflow Datasets website. It features 5.000 total images of 150×150 pixels (RGB), equally divided into 8 categories:



3. Initial Model

Both models are based on the well-known VGG16 architecture.

The initial model is made up of the following architecture:


vgg16_input: InputLayer | input: [(?, 150, 150, 3)] | output: [(?, 150, 150, 3)]

vgg16: Functional | input: (?, 150, 150, 3) | output: (?, 4, 4, 512)

flatten: Flatten | input: (?, 4, 4, 512) | output: (?, 8192)

dense: Dense | input: (?, 8192) | output: (?, 64)

dense_1: Dense | input: (?, 64) | output: (?, 8)

Figure 1: *Diagram of initial model architecture*

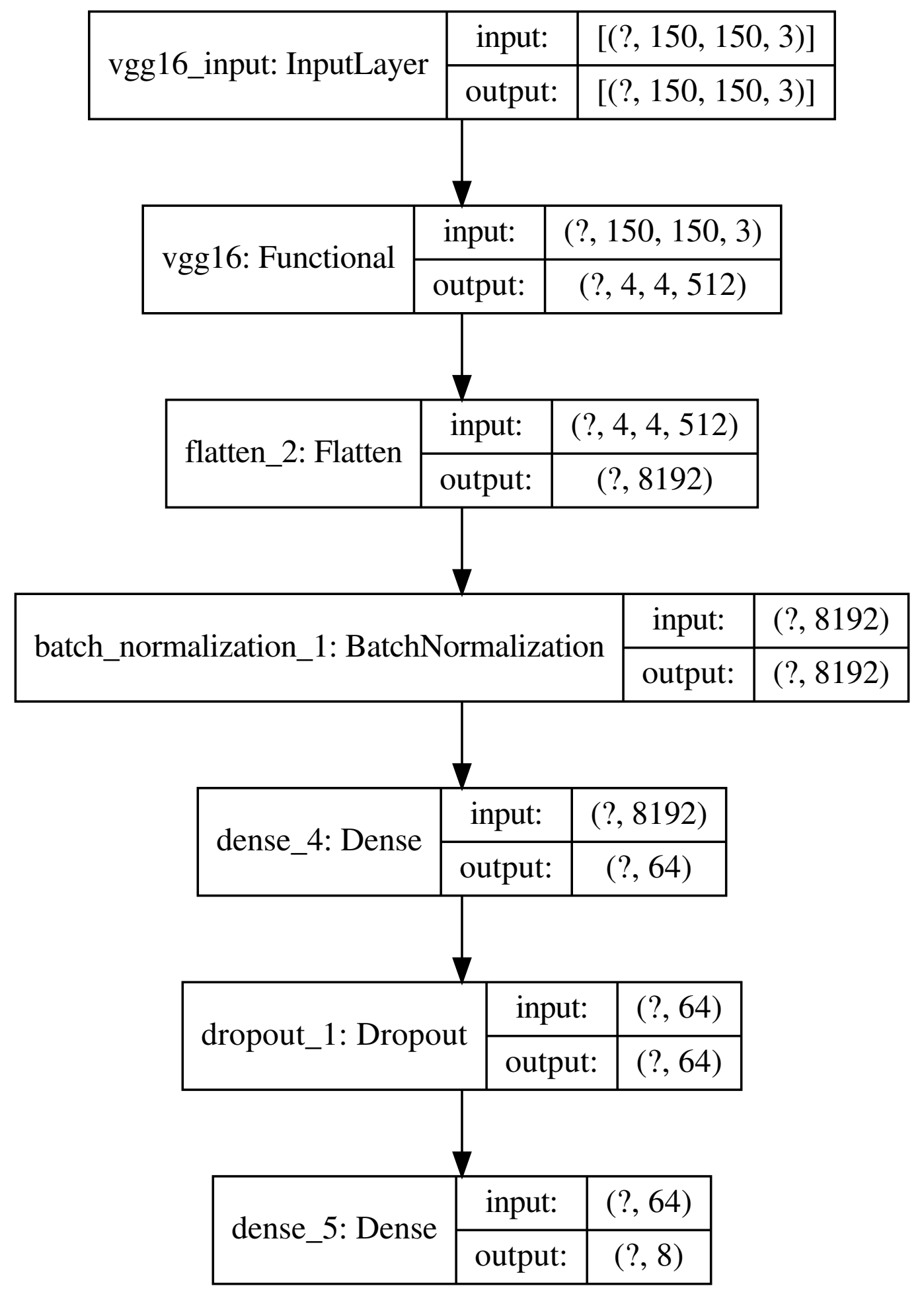
Basically, it's a variation of the VGG16 model, with a 64 neuron Dense layer before the output layer.

This model was trained from scratch on 4.440 images (88,8% of the data) with the Adam (Adaptive Moment Estimation) optimiser for 20 epochs with a batch size of 32 and scored approximately 80% accuracy (with the remaining 11,2% of the images (560) as the validation set).

4. Modified Model

In order to make the model converge even faster and perhaps have even better accuracy, I modified the model to incorporate data augmentation and batch normalisation, while adding a Dropout layer before the output layer.

This model was trained by transfer learning on 4.440 images (88,8% of the data) with imagenet weights, with the Adam (Adaptive Moment Estimation) optimiser for 20 epochs with a batch size of 32 and scored approximately 90% accuracy (with the remaining 11,2% of the images (560) as the validation set).


vgg16_input: InputLayer | input: [(?, 150, 150, 3)] | output: [(?, 150, 150, 3)]

vgg16: Functional | input: (?, 150, 150, 3) | output: (?, 4, 4, 512)

flatten_2: Flatten | input: (?, 4, 4, 512) | output: (?, 8192)

batch_normalization_1: BatchNormalization | input: (?, 8192) | output: (?, 8192)

dense_4: Dense | input: (?, 8192) | output: (?, 64)

dropout_1: Dropout | input: (?, 64) | output: (?, 64)

dense_5: Dense | input: (?, 64) | output: (?, 8)

Figure 2: *Diagram of modified model architecture*

5. Findings

Because the modified model has a better accuracy rate, I chose to use it. Findings were calculated exactly like in prior excercises:

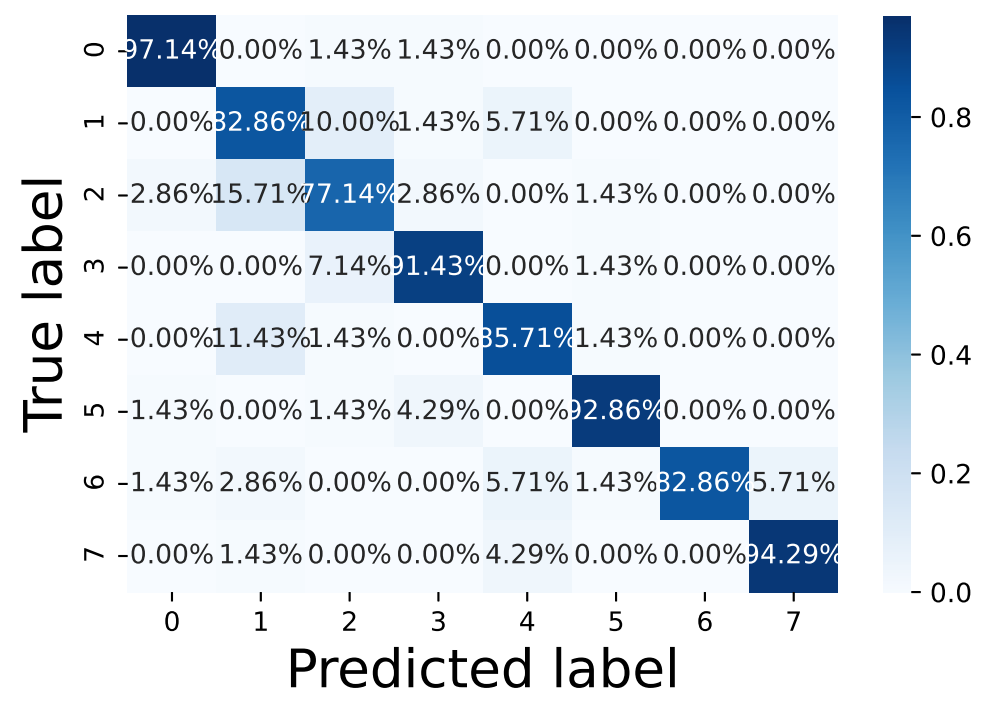


Figure 3: *Modified model's confusion matrix*

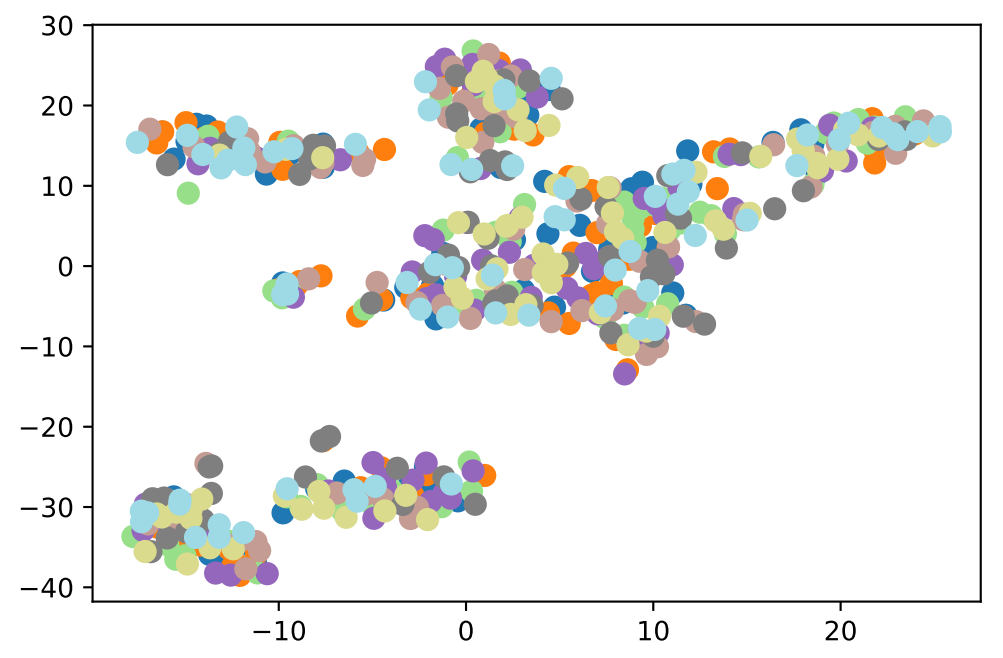


Figure 4: *Modified model's features vector*

6. Results

We feed the following image of 5.000×5.000 pixels (RGB) into the modified model:

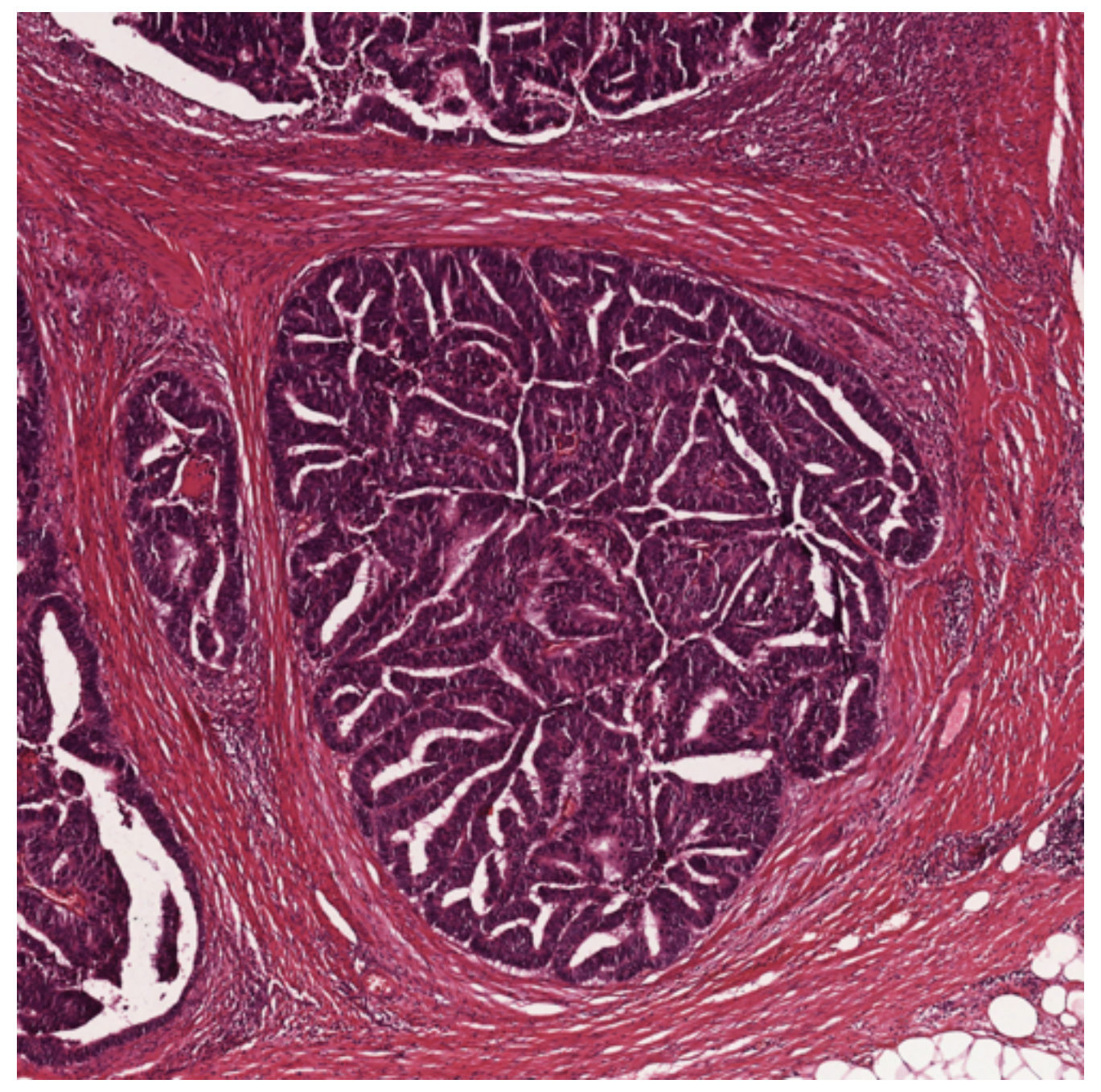


Figure 5: *Large biopsy example image from the colorectal_histology_large dataset*

In order for this image to be accepted by the model, it needs to be sliced into a grid with tiles of 150×150 pixels (RGB). I get a model prediction for each tile, and plot it with different colours for each category next to the original image:

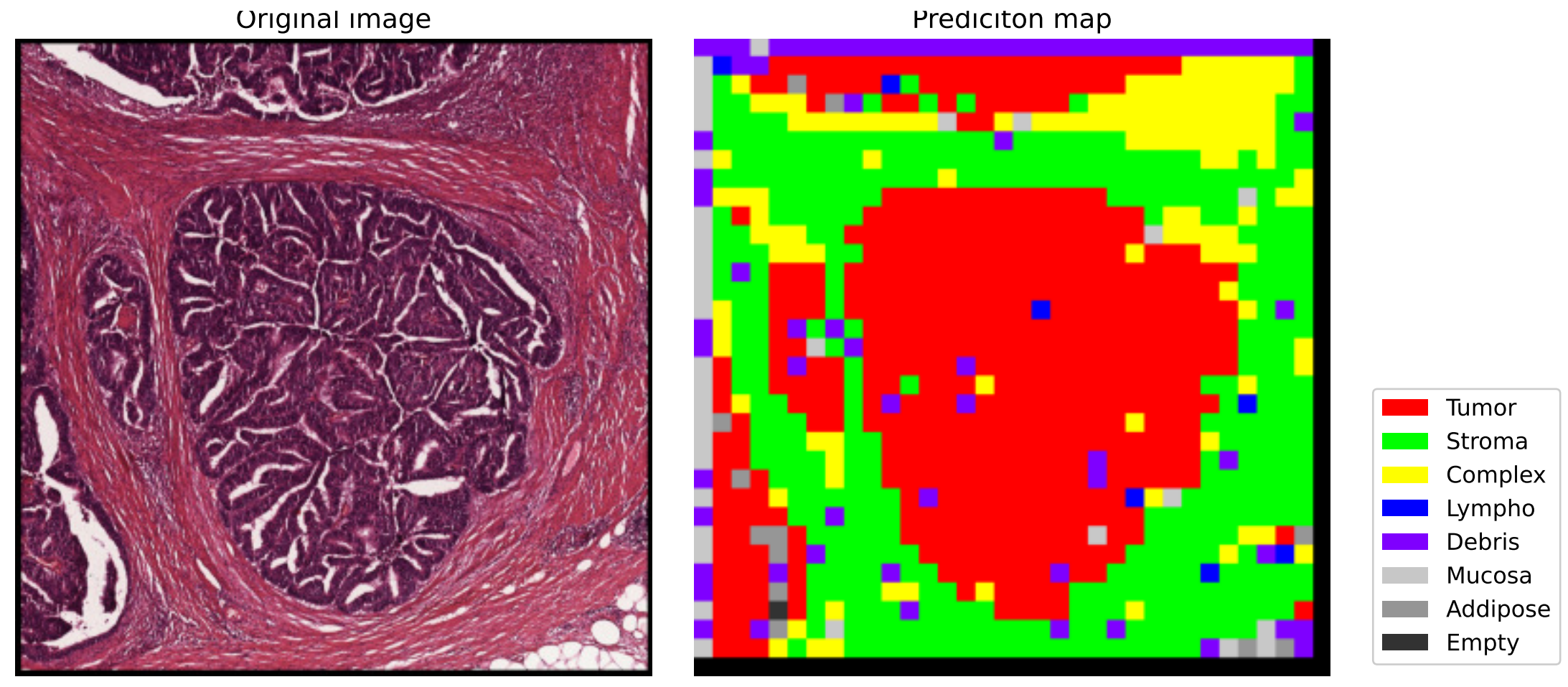


Figure 6: *Colourmap image output after passing through the model*

I performed a similar process to find the "tumor certainty" of each grid tile: I got the "tumor" prediction for each tile, and then I plot it as a heatmap:

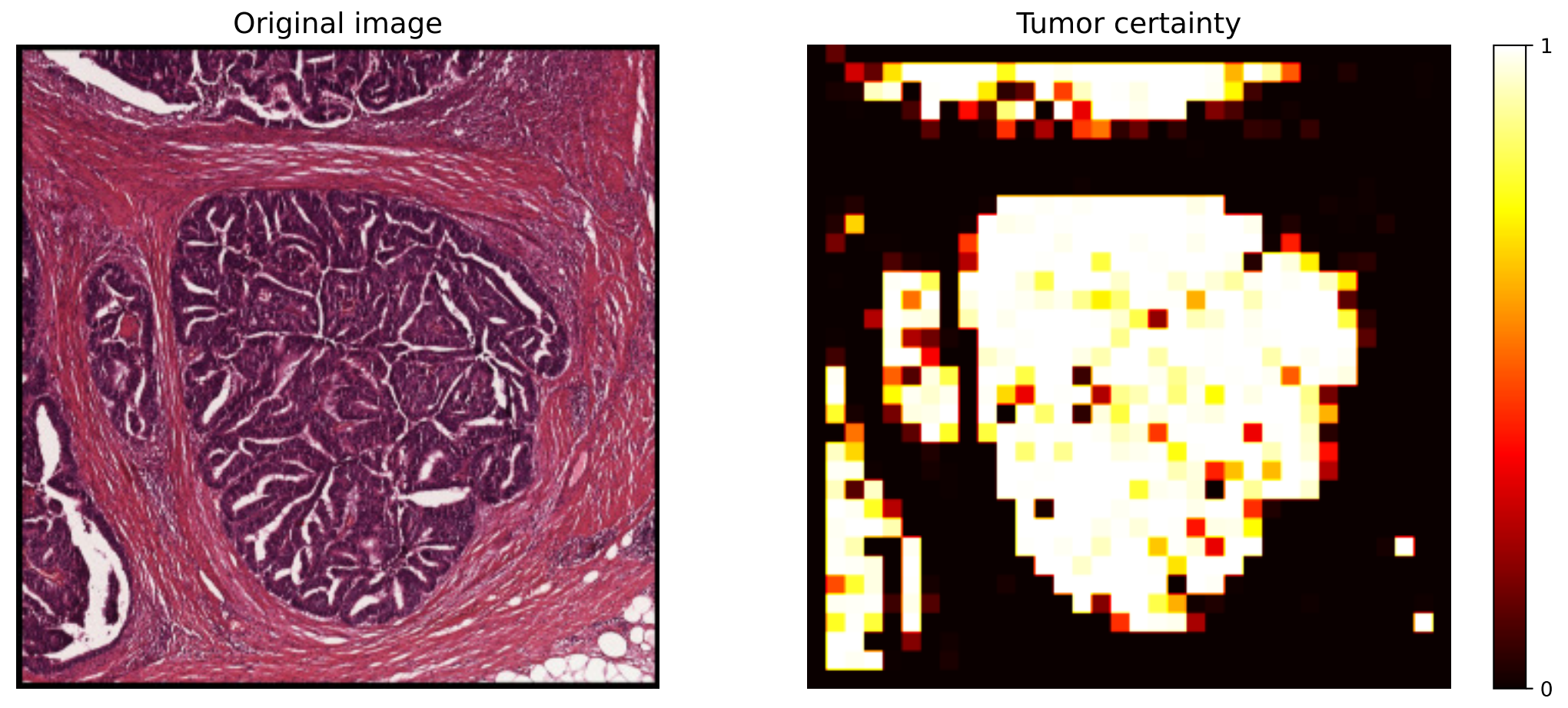


Figure 7: *Heatmap image output after passing through the model*

7. Conclusions

1. Performing transfer learning helps achieve faster model convergence with adequate accuracy.
2. Data generation and batch normalisation helps generalise the data and thus also helps achieve faster model convergence with adequate accuracy.

8. References

1. Deep Learning Course Excercises and Class Work.
2. *colorectal_histology* Dataset Page on Tensorflow Datasets (https://www.tensorflow.org/datasets/catalog/colorectal_histology).
3. *colorectal_histology_large* Dataset Page on Tensorflow Datasets (https://www.tensorflow.org/datasets/catalog/colorectal_histology_large).
4. Tensorflow Documentation (https://www.tensorflow.org/api_docs/python/tf).
5. Keras Documentation (https://keras.io/api/).
6. NumPy Documentation (https://numpy.org/doc/).
7. Matplotlib Documentation (https://matplotlib.org/stable/contents.html).