Contextual Melanoma classification in skin lesion using a Transformer

Reina Yuval, Chikishev Zahar

Abstract- In recent years deep learning techniques have showed promising results in various medical classification challenges, such as skin lesion classification. Usually in these challenges the classification is done on each lesion alone by possessing the lesion's image with some additional metadata. In this paper we present a two stage deep neural network method which uses contextual information from multiple images of the same patient to improve the classification of each lesion. The context is learned by a Transformer network which was originally invented for NLP tasks.

I. Introduction

This document will describe the approach used by team "Yuval and nosound" in Kaggle's competition "SIIM-ISIC Melanoma Classification" [1] in 2020. In this competition the task was to identify melanoma in skin lesion images. One of the tasks in this competition was to use contextual information from multiple images of the same patient to improve the lesion's classification. In this document we will describe a two stages model which first uses a standard classification network for single lesions, and then uses this network as a feature extraction block for the 2nd network which is a transformer that gains contextual knowledge form multiple images and uses it to improve the classification.

In the next section we will give a short introduction of the data used. Section III will present the model. Section IV will describe the training and inference details, and the results will be presented if section V.

Unfortunately, the variance in the results is too high and we weren't able to get conclusive evidence of the improvement.

II. Data

The SIIM - ISIC 2020 competition's data consists of 33126 records, each has a lesion's image and metadata. The metadata includes the following information: patients id, sex, age, the anatomic site, the lesion diagnosis ('nevus', 'melanoma', 'seborrheic keratosis', etc.), is the lesion benign or malignant, and a target – 1 for melanoma and 0 for others. The test set has 10982 records with the same metadata, except diagnosis benign/malignant and target.

There are 2056 different patients in the training data and 690 in the test data. There are 584 (1.76% !!) positive targets in the training data, and about 21% of the patient's have at least one melanomial lesion. The number of images per patient ranges between 2 – 115 in the train data and between 3 – 240 in the test data.

Due to the very low number of positive examples in the train data we also used data from ISIC 2019 competition [3]. In this dataset there are 25331 records and 4522 (17.85%) of them are positive for melanoma. In this dataset the patient id isn't specified, and it was impossible to group records that are related to the same patient.

III. Model

A. *Full model*

Our solution is based on a two steps model:

- Base model for feature extraction per image
- Transformer model – learning from context by combining all the output features from a patient and predict per image.

The inspiration for this kind of model came from a previous Kaggle competition - *RSNA Intracranial Hemorrhage Detection [4]*. In that competition, all the top solutions used a two stages network approach (although none of them used transformers).

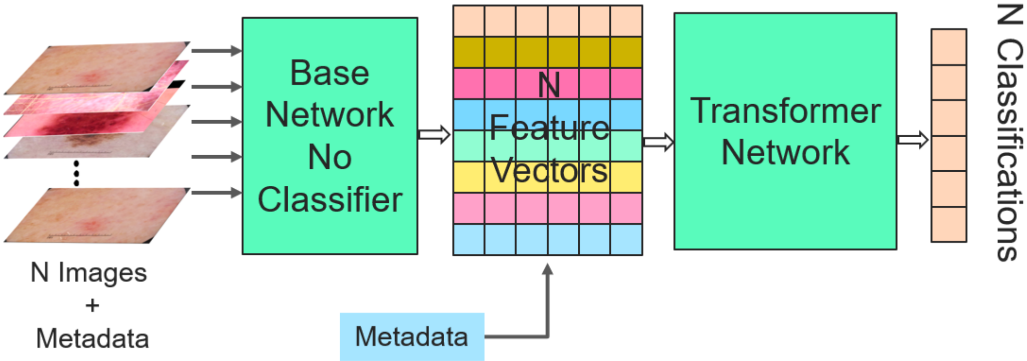

Figure 1 - Full Model

B. Features extraction stage

The feature extraction is done using a “Base Model”. As base model we used models from the EfficientNet [5] family (b3 – b7). All models were pre-trained on Imagenet using noisy student algorithm. The models and weights are from gen-efficientnet-pytorch [6].

The input to the base model is the image and metadata such as age, sex, and anatomic Site. The metadata is processed by a small fully connected network and its output is concatenated to the input of the classification layer of the original EfficientNet network. This vector is going through a linear layer with output size of 256 to create the “features”, and then after an activation layer to the final linear classification layer.

This network has 8 outputs and tries to classify the diagnosis label (there are more than 8 possible diagnoses, but some don’t have enough examples).

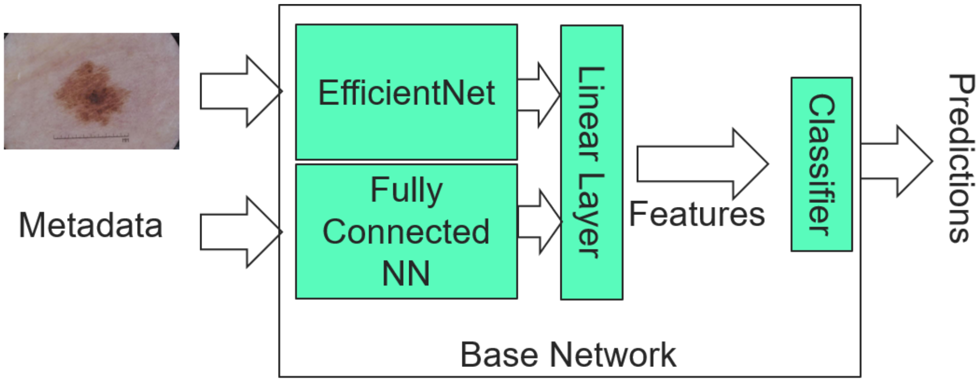

Figure 2 - Feature extraction stage

A. *C. Transformer stage*

The input to the Transformer model is a stack of features from all images belonging to the same patient + the metadata for these images.

The transformer is a stack of 4 transformer encoder layers with self-attention as described in *Attention Is All You Need* [2]. Each transformer encoder layer uses 4 self-attention heads.

The output of the transformer is N*C matrix, where N is the number of feature vectors (the number of images) and C is the number of classes (8 in this case). Hence, **the transformer predicts the class of each feature vector simultaneously, using the information from all other feature vectors**.

The metadata is added using a “transformer style”, i.e. each parameter is transformed to a vector (size 256) using an embedding matrix and then added to the feature vector. For continuous values (like age) the embedding matrix was replaced by a 2 layer fully connected network.

A transformer can be trained on different number of feature vectors, by using padding. But when the range of numbers is very large (2 – 215 in our case), this may cause some implementation issues (like in calculating the loss). To simplify these issues, we limited N to 24 feature vectors, and for each patient we randomly divided the images to groups of size up to 24. This might degrade the prediction as the most “similar” images might accidentally fall into different groups, but as we use TTA, this issue is almost solved.

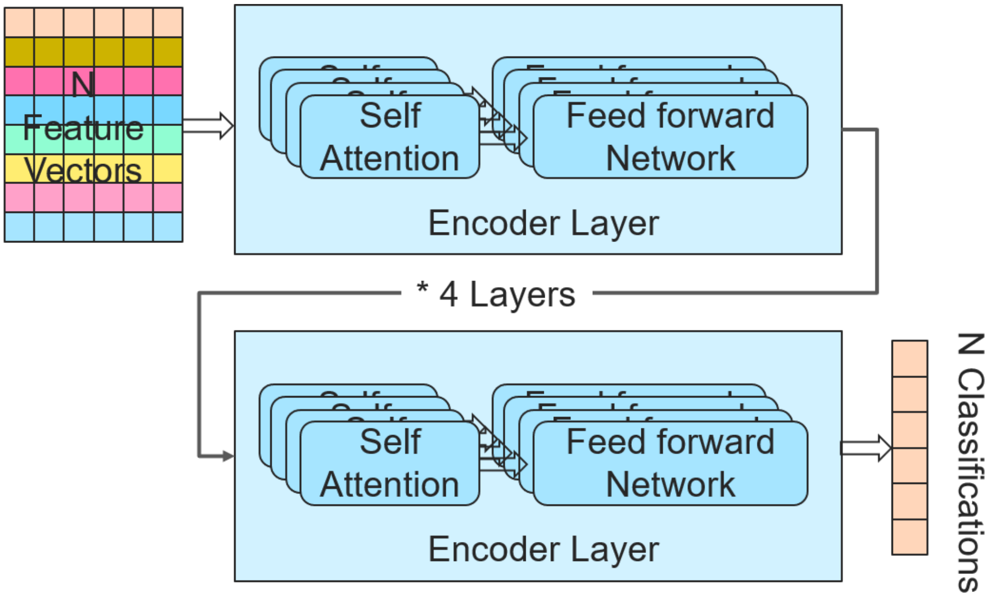

Figure 3 - Transformer model

IV. Training and Inference

A. *Preprocessing*

All images were resized to an aspect ratio of 1:1.5, which was the most popular aspect ratio of the images in the original dataset. We prepared 3 image datasets of sizes 300*450, 400*600, 600*900. Most of the models were trained using the 400*600 dataset, as it balanced between accuracy and speed.

The metadata was coded to integer and a few rare diagnoses were replaced by "unknown" diagnosis.

B. Augmentation

The following augmentations where used while training and inference:

- Random resize + crop
- Random rotation
- Random flip
- Random color jitter (brightness, contrast, saturation, hue)
- Cutout [7] - erasing a small rectangle in the image
- Hair - Randomly adding "hair like" lines to the image
- Metadata augmentation - adding random noise to the metadata as was done in the 1st place solution in ISIC 2019 challenge [8].

C. *Base model training and Inference*

1. *folds*

We split the training data to 3 folds keeping all the images from the same patient in the same fold and making sure each fold has a similar number of patients with melanoma. The ISIC2019's data was also split evenly between the folds. The same folds were kept for the base and the transformer models. To get more diversity we had 3 different splits using 3 seeds

2. *Loss*

The loss function was cross entropy. Although the task is to predict only melanoma, we found it is better to predict the diagnosis which had 8 different classes, one of which was melanoma. The final prediction was the value for the melanoma class, after a softmax function on all classes. We also tried a binary cross entropy on the melanoma class alone and a combination between the two, but using cross entropy gave the best results.

3. *Inference*

For inference each image was augmented differently 16 times and the final prediction was the average. These augmentations were also used for extracting 16 different features vectors per test image. The same was done to extract 12 features vectors for the train images (12 and not 16 because of time limits).

D. Transformer training and inference

We used the same folds for the transformer stage, and the same loss. We also tried using focal loss, but it didn't have a positive effect on the results (we kept it in one of the models of the final ensemble).

The transformer model was trained in two steps. For the first step we used the data from both competitions (2019, 2020). For the 2019 competition we don't have information about the patient, and each image got a different dummy patient id, meaning the transformer didn't have any context to learn from in these images.

In the 2nd stage we fine-tuned the transformer using only the 2020 competition's data.

In both steps we used a sampler that over sampled the larger groups.

In the training and inference steps of the transformer model we randomly choose one of these vectors. Another augmentation is the random grouping as stated above.

TTA was also used in the transformer's inference.

E. Ensembling

The ensemble was done by averaging the outputs before Softmax.

V. Results

Models Ensembled	Private LB	Public LB	Remarks
A. EfficientNet B3 noisy student image size 400*600 B. EfficientNet B4 noisy student image size 400*600 C. EfficientNet B5 noisy student image size 400*600 D. EfficientNet B6 noisy student image size 600*900 E. EfficientNet B7 noisy student image size 400*600	**0.9441**	**0.9575**	Without context
• All the "without context" model + • Transformer on features from A. • Transformer on features from B. • Transformer on features from C using focal loss • Transformer on features from D. • Transformer on features from E.	**0.9430**	**0.9582**	With context
Transformer on features from B5 noisy student image size 400*600	**0.9397**	**0.9602**	Best Single model (Public LB)

Although there are some signs that using the transformer to learn from the contextual information, improve the classification, the differences are too small compared to the noise (measured by the standard deviation between the score for different folds), and we can't show any conclusive evidence of improvement.

VI. Discussion

In this paper we demonstrated how transformers can help learn contextual information in a skin lesion classification challenge. This approach can be used in other challenges where contextual information is available.

Transformers can have higher impact in a real-world lesion classification scenario when previous images of a patient are already tagged, using this information to predict the current tagging, makes the model a seq2seq model – which is what a full transformer (encoder +decoder) was invented for.

VII. Acknowledgment

We wanted to thank the Kaggle team and the competition organizers from SIIM and ISIC for giving us the opportunity to work on this interesting and important challenge.

VIII. References

1. https://www.kaggle.com/c/siim-isic-melanoma-classification/overview. Kaggle. [Online]
2. Vaswani, N. Shazeer, N Parmar, J Uszkoreit, L Jones, A. N. Gomez, Ł. Kaiser, I Polosukhin. Attention Is All You Need. arXiv:1706.03762v5. [Online] https://arxiv.org/pdf/1706.03762v5.
3. https://challenge2019.isic-archive.com/. [Online] 2019.
4. https://www.kaggle.com/c/rsna-intracranial-hemorrhage-detection. Kaggle. [Online] 2019.

5. Le, Mingxing Tan and Quoc V. EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks. arxiv.org. [Online] 2019. https://arxiv.org/abs/1905.11946.
6. Wightman, Ross. https://github.com/rwightman/gen-efficientnet-pytorch. Github. [Online] 2020.
7. Taylor, Terrance DeVries and Graham W. Improved Regularization of Convolutional Neural Networks with Cutout. arxiv.org. [Online] 2017. https://arxiv.org/abs/1708.04552.
8. Nils Gessert, Maximilian Nielsen,Mohsin Shaikh, René Werner, Alexander Schlaefer. Skin lesion classification using ensembles of multi-resolution EfficientNets with meta data. [Online] 2020. https://doi.org/10.1016/j.mex.2020.100864.

IX. Authors

Yuval Reina – BScEE , MBA , Tel - Aviv, Israel, Yuval.Reina@gmail.com
Zahar Chikishev– M.Sc. in applied math, Berlin, Germany, zahar.chikishev@gmail.com