Using Soft-Attention Layer to Improve FaceNet and Brain Tumor Classification Performance

Soroush Naseri
Undergraduate Computer Engineering Student
Ferdowsi University of Mashhad

Modjtaba Rouhani
Associate Professor of Computer Engineering
Ferdowsi University of Mashhad

I. ABSTRACT

In various applications, including clinical and face detection, neural networks need to prioritize and emphasize the most crucial aspects of input images. The Soft-Attention mechanism facilitates this objective by enabling neural networks to discern and enhance important features while suppressing noise-inducing ones. This project aims to assess the effectiveness of Soft-Attention within deep neural architectures. Initially, we integrate this mechanism into DenseNet to detect brain tumor conditions. Subsequently, we apply the Soft-Attention layer to FaceNet in conjunction with ResNet50. The concept for our project stemmed from an essay titled "Soft-Attention Improves Skin Cancer Classification Performance." This essay focused on enhancing the accuracy of classifying skin cancer into three distinct categories.

II. INTRODUCTION

Brain tumors pose a grave threat to health due to their potential to disrupt critical brain functions. These abnormal growths can lead to severe neurological symptoms, including headaches, seizures, and cognitive impairment. The danger lies in their ability to exert pressure on vital brain structures, causing further damage and impairing essential bodily functions. Malignant brain tumors, in particular, carry a high risk of aggressive growth and metastasis, leading to devastating consequences for affected individuals. Given their life-threatening nature, prompt diagnosis and intervention are crucial in mitigating the dangers posed by brain tumors and improving patient outcomes.

Various neural network architectures can be employed to classify brain images into three classes: meningioma, glioma, and pituitary tumor. In this study, we utilize the DenseNet architecture and conduct experiments with and without the Soft Attention layer to evaluate its impact on classification performance. By comparing the results obtained from these two cases, we aim to assess the effectiveness of Soft Attention in improving the classification accuracy of brain tumor types.

For another application Face detection holds significant importance across various fields such as security, surveillance, entertainment, and human-computer interaction. In security and surveillance, accurate face detection systems aid in identifying individuals for law enforcement purposes, enhancing public safety, and monitoring crowded areas for potential threats. In entertainment, face detection technology is used for personalized advertising, augmented reality filters, and facial recognition in video games. Moreover, in human-computer interaction, face detection enables natural user interfaces, emotion recognition, and personalized user experiences. Overall, the importance of face detection lies in its multifaceted applications that contribute to safety, entertainment, and improved interaction between humans and technology.

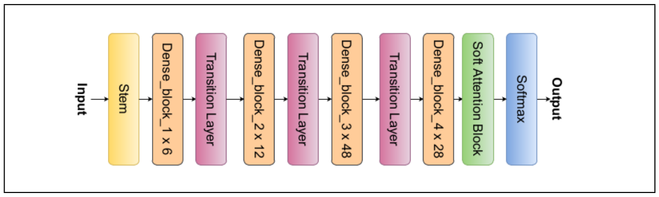

Fig. 1. End to end schema of DenseNet201 with Soft Atten-tion Block.[16].

FaceNet, developed by Google in 2016, is a neural network designed to generate a vector representation for each face. This vector representation enables images with similar faces to have vectors that are close together in Euclidean space. To further enhance its performance and reduce losses, Soft Attention is integrated into its deep architecture. This addition aims to improve the network's ability to focus on relevant facial features, ultimately enhancing its face recognition capabilities.

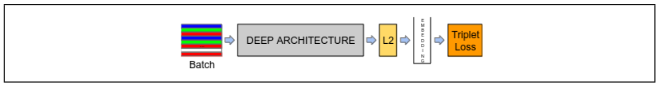

Fig. 2. Model structure. Our network consists of a batch in- put layer and a deep CNN followed by L2 normalization, which results in the face embedding. This is followed by the triplet loss during training.

This research investigates the impact of incorporating soft attention mechanisms into deep neural networks. Deep learning architectures discern image classes by learning significant features and nonlinear relationships. The inclusion of soft attention enhances performance by prioritizing relevant regions of the input. Furthermore, the transparency provided by the soft attention mechanism renders the image classification process understandable to medical practitioners, as it delineates the portions of the input utilized by the network for classification, thereby fostering confidence in the classification model.

III. BRAIN TUMOR CLASSIFICATION

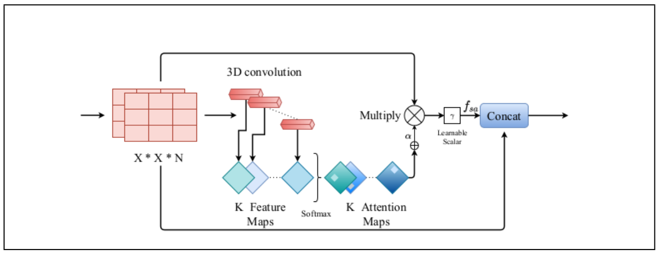

Fig. 3. The schema for Soft Attention Block

A. Brain Tumor Dataset

In this section, we utilized a dataset comprising 3064 T1-weighted contrast-enhanced images sourced from 233 patients, encompassing three distinct types of brain tumors: meningioma (708 slices), glioma (1426 slices), and pituitary tumor (930 slices). To ensure uniformity, all images were resized to dimensions of 224x224. Subsequently, the dataset underwent a thorough cleaning process to rectify class imbalances. This involved employing both over-sampling and under-sampling techniques to ensure an equal distribution of images across all classes. Additionally, to standardize the data, pixel values were normalized by dividing each pixel by 255, thus constraining the pixel values within the range of 0 to 1.

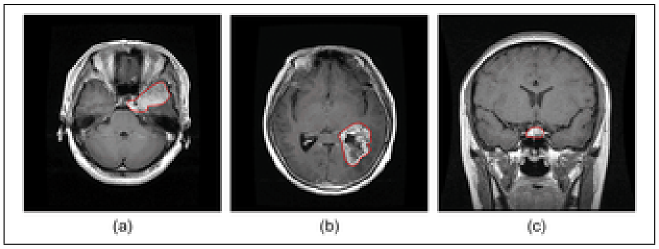

Fig. 4. Brain tumor dataset sample

B. Architecture

Drawing inspiration from the research conducted by Xu et al. [28] on image caption generation and the work by Shaikh et al. [18], which utilized attention mechanisms for handwriting verification, this paper explores the application of soft attention in skin cancer classification.

The soft attention module, as described in papers [18] and [23], utilizes the feature tensor propagated through the deep neural network as its input.

$$fsa = \sum_{k=1}^{K} t(softmax(W^k * t))$$

The feature tensor $t \in R^{h \times w \times d}$ serves as input to a 3D convolutional layer [24] with weights $W_k \in R^{h \times w \times d \times K}$, where K denotes the number of 3D weights. This convolutional output undergoes normalization via the softmax function to generate $K = 16$ attention maps. These attention maps, depicted in Figure 3, are combined to generate a unified attention map, acting as a weighting function denoted as α. Subsequently, α is multiplied by t to selectively amplify the significant feature values, further adjusted by γ, a trainable scalar. Finally, the attentively scaled features (f_{sa}) are merged with the original feature t via a residual branch. Throughout training, γ is initialized at 0.01 to facilitate gradual learning for regulating the attentional requirements of the network.

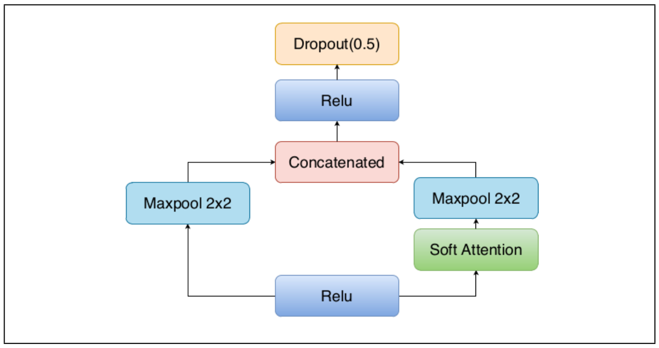

Fig. 5. The schema for Soft Attention Block

In DenseNet201[8], the soft attention layer is incorporated into the 4th dense block, operating on a feature map size of 7 x 7, as illustrated in Figure[1]. Similar to the previous model, the integration of the soft attention layer follows the procedure used in the Inception ResNet V2[22] architecture.[4].

C. Loss Function

In this experiment, with three distinct classes of brain tumor, the categorical cross-entropy loss (LCCE) is employed to optimize the neural network.

$$LCCE = -\sum_{i=1}^{C} t_i \log(\sigma(z)_i)$$

with :

$$\sigma(z)_i = \frac{e^{z_i}}{\sum_{j=1}^{C} e^{z_j}}$$

D. Metric

- Accuracy measures the overall correctness of a model's predictions across all classes.
- It calculates the ratio of correctly predicted instances (true positives and true negatives) to the total number of instances in the dataset.
- The formula for accuracy is:

$$Accuracy = \frac{NumberofCorrectPredictions}{TotalNumberofPredictions}$$

- Recall, also known as sensitivity or true positive rate, measures the ability of a model to correctly identify all relevant instances of a specific class.
- It calculates the ratio of true positive predictions to the total number of actual positive instances in the dataset.
- The formula for recall is:

$$Recall = \frac{TruePositives}{TruePositives + FalseNegatives}$$

  where:
  - True Positives (TP) are the instances that are correctly predicted as positive.
  - False Negatives (FN) are the instances that are actually positive but incorrectly predicted as negative.

Macro average calculates the average performance across all classes independently. It computes the metric (such as precision, recall, or F1-score) for each class separately and then takes the average of these scores. Each class is given equal weight in the calculation, regardless of class size. The formula for macro average is:

$$MacroAvr = \frac{1}{N}\sum_{i=1}^{N} Metric_i$$

where N is the number of classes and $Metric_i$ is the metric value for class i.

Micro average calculates the aggregate performance across all classes by considering the total number of true positives, false positives, and false negatives. It computes the metric (such as precision, recall, or F1-score) using the total counts of true positives, false positives, and false negatives for all classes combined. Each instance is given equal weight in the calculation, regardless of class. The formula for micro average is:

$$MicroAvr = \frac{TotalTruePositives}{TotalTruePositives + TotalFalsePositives}$$

$$= \frac{TotalTruePositives}{TotalPredictedPositives}$$

AUC is a metric used to evaluate binary classification models based on their ROC (Receiver Operating Characteristic) curve. ROC curve is a graphical plot that illustrates the performance of a binary classifier across different thresholds. AUC represents the area under the ROC curve, indicating the model's ability to distinguish between positive and negative classes. AUC value ranges from 0 to 1, where 0.5 represents random guessing and 1 represents perfect classification. Higher AUC values indicate better classifier performance. AUC can be calculated using various methods, such as the trapezoidal rule or the Wilcoxon-Mann-Whitney statistic.

E. Results

For DensNet without attention layer:

$$Accuracy : 0.984$$

	Micro	Macro
Percision	0.984	0.980
Recall	0.984	0.984

The ROC AUC score of First class: 0.998
The ROC AUC score of second class: 0.999
The ROC AUC score of third class: 0.999

For DensNet with attention layer:

$$Accuracy : 0.973$$

	Micro	Macro
Percision	0.973	0.973
Recall	0.973	0.966

The ROC AUC score of First class: 0.996
The ROC AUC score of second class: 0.998
The ROC AUC score of third class: 0.999

IV. FaceNet Architecture

A. Method

The FaceNet discusses, a deep convolutional network, employing two core architectures: Zeiler and Fergus style networks and recent Inception type networks. The key aspect of the approach lies in end-to-end learning using triplet loss, aiming for an embedding f(x) from an image x into a feature space Rd. This embedding minimizes the squared distance between faces of the same identity and maximizes it between faces of different identities. While not directly compared to other losses, such as pairs of positives and negatives, the triplet loss is considered more suitable for face verification as it enforces a margin between each pair of faces from one person to all others, allowing faces for one identity to live on a manifold while maintaining discriminability .

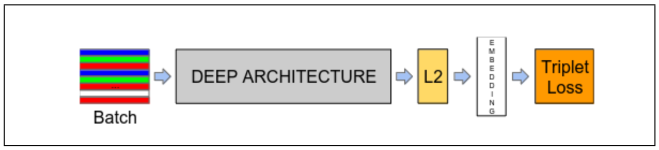

Fig. 6. FaceNet block

B. Method

The architecture utilized in this study is based on FaceNet, employing a deep network. However, a pretrained ResNet on the WebFace dataset (Fig. 7) was used. Additionally, the isgrad parameter of all layers was set to false, allowing only the gradient of the last layer to change. A Soft Attention block was incorporated after the deep network and before L2 normalization. This setup enables the improvement of weights in both the last layer of ResNet and the Soft Attention layer.

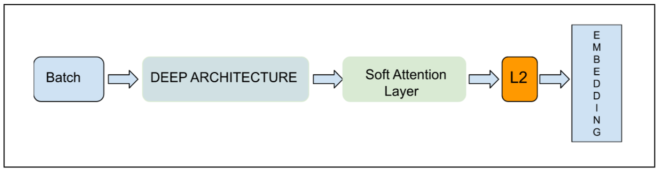

Fig. 7. Modified FaceNet block

Following the Soft Attention layer, L2 normalization is applied. Subsequently, a fully connected layer produces a vector with 128 dimensions. This process allows the creation of a 128-dimensional vector for each face.(fig7)

C. Triplet Loss

Triplet loss is a concept used in deep learning, particularly in tasks like face recognition and verification. The goal of triplet loss is to learn a feature representation for each input such that similar examples are closer together in the feature space, while dissimilar examples are farther apart.

In triplet loss, each training example consists of three images: an anchor image (representing the ”positive” example), a positive image (another image of the same class as the anchor), and a negative image (an image from a different class). The loss function penalizes the model if the distance between the anchor and the positive image is not smaller than the distance between the anchor and the negative image by a predefined margin.

Mathematically, given embeddings of the anchor A, positive P, and negative N images as $f(A)$, $f(P)$, and $f(N)$ respectively, the triplet loss L is computed as:

$$L = \max\{d(f(A), f(P)) - d(f(A), f(N)) + \alpha, 0\}$$

where $d(\cdot, \cdot)$ represents a distance metric (e.g., Euclidean distance or cosine similarity), and α is a margin that defines the minimum difference required between the distances of positive and negative pairs.

By optimizing the triplet loss, the model learns to map similar images closer together in the feature space while pushing dissimilar images farther apart, ultimately leading to improved performance in tasks like face verification and recognition.

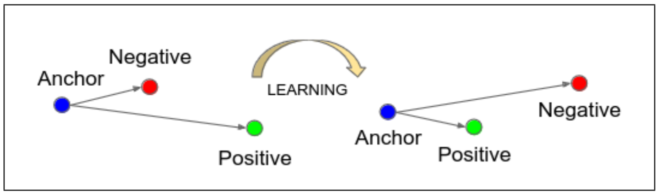

Fig. 8. The Triplet Loss minimizes the distance between an an- chor and a positive, both of which have the same identity, and maximizes the distance between the anchor and a negative of a different identity.

D. Dataset

The first dataset I utilized for pretraining my ResNet network is CASIA-WebFace. Employing transfer learning, I leveraged this dataset to initialize the network’s weights before fine-tuning it with the VGG-Face2 dataset to train the remaining layers. This approach allows for the efficient transfer of knowledge from the pretraining dataset to the target dataset, enhancing the network’s ability to learn discriminative features specific to face recognition tasks. CASIA-WebFace, known for its large-scale collection of facial images sourced from the internet, provides a diverse and extensive dataset suitable for training deep learning models for face-related tasks.

The WebFace dataset, sometimes referred to as WebCasino Face dataset, is a large-scale face dataset commonly used for training and evaluating face recognition algorithms. It contains a vast collection of facial images sourced from the internet, capturing a wide range of poses, expressions, and lighting conditions. The dataset is known for its diversity and scale, making it suitable for training deep learning models for various face-related tasks.The dataset contains 494,414 face images of 10,575 real identities collected from the web.(fig 9)

The VGGFace2 dataset is made of around 3.31 million images divided into 9131 classes, each representing a different person identity. The dataset is divided into two splits, one for the training and one for test. The latter contains around 170000 images divided into 500 identities while all the other images belong to the remaining 8631 classes available for training. While constructing the datasets, the authors focused their efforts on reaching a very low label noise and a high pose and age diversity thus, making the VGGFace2 dataset a suitable choice to train state-of-the-art deep learning models on face-related tasks. The images of the training set have an average resolution of 137x180 pixels, with less than 1 percent at a resolution below 32 pixels (considering the shortest side).(fig 9)

Fig. 9. VGGFace2 dataset

E. Result

This network outputs 128-dimensional vectors for each face. Ideally, faces corresponding to the same person should be closer in this vector space compared to faces of different individuals. This objective aligns with the previously introduced Triplet Loss function. During training, positive and negative samples for each ”anchor” face are randomly chosen