Visualization and Analysis of Transformer Attention*

Salvatore Calderaro[1], Giosué Lo Bosco[1], Riccardo Rizzo[2] and Filippo Vella[2,*]

[1]Department of Mathematics and Computer Science, Università degli Studi di Palermo, via Archirafi 34, Palermo, Italy
[2]Institute of High Performance Computing and Networking, National Research Council of Italy CNR, via Ugo La Malfa 153, Palermo, Italy

Abstract

The capability to select the relevant portion of the input is a key feature to limit the sensory input and focus on the most informative collected part. The transformer architecture is among the most performing deep neural network architectures due to the attention mechanism. The attention allows us to spot relevant connections between portions of the images and highlight these connections. Since the model is complex, it is not easy to determine which are these connections and the important areas. We discuss a technique to show these areas and highlight the regions most relevant for label attribution.

1. Introduction

In 1890, the psychologist and philosopher William James, in his book "The Principles of Psychology", wrote that attention, "is the taking possession by the mind, in clear and vivid form, of one out of what may seem several simultaneously possible objects or trains of thought. It implies withdrawal from some things to deal effectively with others." [1]. For the visual aspect, we can consider that attention deals with focusing on a portion of the visual input and selecting specific areas.

In 2001, Itti and Koch modelled visual attention, choosing salient areas in the image. They search for visual attributes such as edges, intensity contrast, corners and junctions, considering that the neurons at the earliest stages of the human visual system detect simple visual attributes. At the same time, higher neural levels are more specialized and detect high-level visual areas such as corners, junctions or real-world objects [2].

The attention mechanisms implement selection procedures focusing on the part of the image according to specific features. The artificial attention models does not follow the biological mechanism of the human brain; they tend to highlight the most salient part of the input according to the pattern they learned in the training phase.

The attention-based system can be considered as composed of three components: A first

1st Workshop on AI for Perception and Artificial Consciousness, AIxIA november 06–09, 2023, Rome, Italy
*Corresponding author.
†These authors contributed equally.
salvatore.calderaro01@unipa.it (S. Calderaro); giosue.lobosco@unipa.it (G. Lo Bosco); riccardo.rizzo@icar.cnr.it (R. Rizzo); filippo.vella@icar.cnr.it (F. Vella)
http://www.pa.icar.cnr.it/vella/ (F. Vella)
0000-0003-0999-6345 (S. Calderaro); 0000-0002-1602-0693 (G. Lo Bosco); 0000-0001-5007-6925 (R. Rizzo); 0000-0002-2502-0062 (F. Vella)

process “reads” raw data (such as words in an input sentence) and converts them into a representation, with one feature vector associated with each word position. A second component stores the reader’s output, and can be considered as a “memory" containing a sequence of facts. A final process “exploits" the content of the memory and performs a sequential task. At each time step, this process can put attention on the content of one or a few memory elements [3] [4].

The representation used in the first process can be derived from an encoder-decoder architecture. The model represents the input in the embedding space and processes all the input items. The list of these representations, coupled with the decoder’s hidden states, is used to select which inputs will be used to generate the output. The input, the previous hidden states and the encoded vectors are used to evaluate scores that indicate how much input aligns with the output. Typically, a softmax is used to normalize the scores and interpret them as weights. The encoded vectors are scaled by the obtained weights and are used to generate a context vector. This context is given to the decoder portion of the architecture and is used to generate the output.

According to Lindsay [5], “This type of artificial attention is thus a form of iterative reweighting. Specifically, it dynamically highlights different components of a pre-processed input as they are needed for output generation. This makes it flexible and context-dependent, like biological attention".

We are interested in “selective” attention, which is the capability to focus on a limited portion of the input, filtering out a huge quantity of details.

The transformers, proposed by Vaswani et al. [6], evaluate an attention function with a neural model, and it was formerly used to form a context for the words to be translated. Here, we consider the evolution of transformers used to process visual input, and we are interested in the visualization of the attention inside the image. In classifying images, the visualization of attention is helpful to highlight the regions of the images that contributed to the production of a given label. According to Vaswani [6], the most significant area for creating the context is letting the model associate the label with the input. Evaluation of the salient and relevant part of the image can be drawn, considering which details are the most informative, and an evaluation of the trust of the classification can be assessed. If the classification is performed with attention to details relevant to the domain’s expert, confidence in the model and its choices increases. If the attention highlights the border of the image or homogeneous areas, it could be inferred that overfitting is present. The correct evaluation and comparison of attention is, therefore, beneficial for assessing machine learning systems. Recent works in explainability [7],[8], [9] and [10] witness the importance of understanding what is relevant to tune and assess the classification results in the machine learning models. Other works are focused on visualization of relevant image areas with transformer’s models, such as [11], [12]. These works are focused on the attention flow across the network layers. We adopt an alternative stance considering that the visualization of the last attention layer is very informative since the activation of the last layer is used to associate the final label to the given input. The next section of the paper describes the vision transformer and the technique we used to visualize attention. In section 4, the experimental part is described, and some results are shown. Conclusions are drawn in section 5.

2. Vision Transformer

The Vision Transformers (ViT) [13] are customized versions of the original Transformers [6]. This architecture is an encoder-decoder designed for word sequence processing and is more accurate than traditional recurrent networks when replicated in layers and piled with a multi-layer perceptron (MLP) layer. It is characterized by the so-called *self-attention layer*. The ViT (see Figure 1) incorporates the transformer's ability to take into account the long-term relationships in the input data. It should be noted that the ViT solely relies on an encoder that incorporates a *multi head* self-attention (MHA) and does not contain a decoder part.

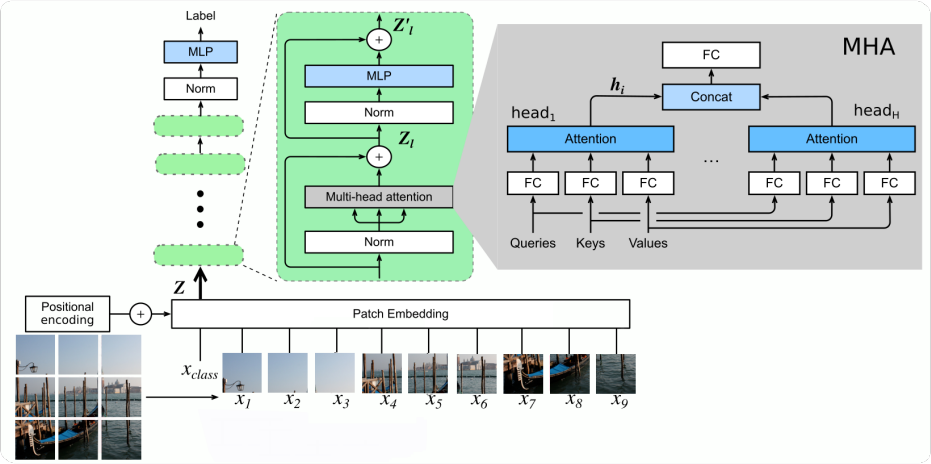


Figure 1: A sketch of the transformer structure.

A tokenized input sequence $Z = [\boldsymbol{z_1}, \ldots, \boldsymbol{z_t}]$ is the input for a single self-attention layer, where $z_i \in \mathbb{R}^m$, is used to compute a hidden representation $[\boldsymbol{h_1}, \ldots, \boldsymbol{h_t}]$ by the following formula:

$$\boldsymbol{h_i} = [\boldsymbol{W_v z_1}, \ldots, \boldsymbol{W_v z_t}] \, \text{softmax}\left[\frac{(\boldsymbol{W_q z_i} \cdot \boldsymbol{W_k z_1})}{\sqrt{d}}, \ldots, \frac{(\boldsymbol{W_q z_i} \cdot \boldsymbol{W_k z_t})}{\sqrt{d}}\right] \tag{1}$$

where $\boldsymbol{W_q}, \boldsymbol{W_k}, \boldsymbol{W_v}$ indicates the *queries, key, and values* matrices, respectively, i.e., the learning parameters of the self-attention layer. The hidden component h_i is the relevance of the token z_i for generating a corresponding target. The MHA is (1) the parallel computation of several single self-attentions, (2) the concatenation of the corresponding hidden representations, and (3) their computation by an MLP. Applying the MHA enables us to derive different kinds of relationships between tokens.

Sequence processing suffers from the self-attention's permutation-invariant property, so a *positional encoding* is used to spatially contextualize a symbol in a sequence with a relative and an absolute position. In particular, a positional encoding vector is appended to each element of the tokenized input sequence $[\boldsymbol{x_1}, \ldots, \boldsymbol{x_t}]$.

In the context of ViT, an image $\mathbf{X}$ with r rows, c columns, and p channels is tiled into a set of $s \times s$ square patches, each one representing a token of the transformer input sequence. The linearization of each p channels $s \times s$ patch is a token $\boldsymbol{x}_i$, and the length of the sequence is set to be $t = \frac{r \times c}{s^2}$. Each layer of the ViT receives and returns vectors of the same dimension d. For this reason, the embedding of the patches must project a linearized patch of length $s^2 \times p$ into a vector of dimension d.

The input sequence of the ViT is $\boldsymbol{Z} = [\boldsymbol{x}_{class}, \boldsymbol{E}\boldsymbol{x}_1, \boldsymbol{E}\boldsymbol{x}_2, \ldots, \boldsymbol{E}\boldsymbol{x}_t] + \boldsymbol{E}_{pos}$, where $\boldsymbol{E} \in \mathbb{R}^{(s^2 \times c) \times d}$ is a learned embedding matrix, and $\boldsymbol{E_{pos}} \in \mathbb{R}^{(t+1) \times d}$ is the positional encoding matrix. This input sequence starts with an additional component $\boldsymbol{x}_{class}$ used to capture a representation of the whole sequence, such as a weighted average of the tokens in the sequence.

The single distinguishing feature of ViT is an encoder portion made up of n encoding blocks joined together, each consisting of an MHA followed by an MLP with one hidden layer. Each of these is subjected to layer normalisation (LN), and an additional residual connection is added at the output. The MLP employs the Gaussian Error Linear Unit activation function. In formulas:

$$\begin{aligned} \mathbf{Z}'_\ell &= \text{MHA}(\text{LN}(\mathbf{Z}_{\ell-1})) + \mathbf{z}_{\ell-1} \\ \mathbf{Z}_\ell &= \text{MLP}(\text{LN}(\mathbf{Z}'_\ell)) + \mathbf{z}'_\ell \end{aligned} \qquad \ell = 1, \ldots, n. \tag{2}$$

Transfer learning is always advantageous and becomes necessary for datasets with few images for the case of image classification by deep models. A pre-trained model on a large dataset is taken into account for the case of ViT, followed by fine-tuning on a particular task using a dataset with fewer examples. MLP is applied during the pre-training and fine-tuning processes. In each of the two cases, $\boldsymbol{z}_n^0$, i.e. the vector corresponding to $\boldsymbol{x}_{class}$ after the n encoder blocks, is used as input. An MLP is pre-trained using a huge dataset, such as ImageNet. Another MLP that returns a vector with the same size as the number of classes is used for fine-tuning.

3. Visualization and Analysis of the Attention

The method discussed here is applied to perform the visualization of the attention.

Since a ViT is composed of numerous concatenated blocks, the visualization of the attention is complex. Considering that the final block has unquestionably the highest level of abstraction, only this block is considered in the visualization paradigm. There are h separate attention heads in each block, and each one evaluates $t + 1$ distinct attentions, one for each patch and one for the token $\boldsymbol{x}_{class}$. In the visualization, we decided to consider the softmax attention s_h of to $\boldsymbol{x}_{class}$ as the query and the embeddings of the patches as the keys, so described:

$$s_h = \text{softmax}\left[\frac{(\boldsymbol{W_q x_{class}} \cdot \boldsymbol{W_k z_1})}{\sqrt{d}}, \ldots, \frac{(\boldsymbol{W_q x_{class}} \cdot \boldsymbol{W_k z_t})}{\sqrt{d}}\right] \tag{3}$$

To obtain a vector S of t components, one associated with each token and subsequently to each patch, one softmax attention vector for each head attention was obtained. To aggregate the softmax attention vector of each head, we use different aggregation functions: the mean and the maximum, and we propose the division between the mean and the standard deviation defined in the following equation:

$$F(Attention) = \frac{mean\left[s_1, s_2, \ldots, s_h\right]}{std\left[s_1, s_2, \ldots, s_h\right]} \tag{4}$$

where h is the number of heads. This function is useful to analyze the contribution of the single heads. While the mean and the maximum typically show the results of one head that has produced a stronger result, the function in eq.4 shows the points where there is the maximum agreement among all the heads. If all the heads produce the same or slightly similar activation values in a given point, the standard deviation is small and the denominator will generate a larger value. If the heads produce, for the same point, values in a large range, the denominator will be low and the final point will be not evident in the final attention map.

To superimpose the created attention map onto the input image to the ViT was created by rearranging these t values in a grid of H/P and W/P and scaling it to a resolution of $H \times W$ (see Figures 2,3,4 for an example). To generate a sufficiently smooth map throughout the resizing process, bilinear interpolation was employed, then a median filter with a structural element of radius $P/2$, followed by a Gaussian blur with $\sigma = P/4$.

4. Experiments and Results

For the experimental activities, we considered the models implemented in the PyTorch Image Models (a.k.a. Timm)[14] involving different kinds of transformers, the "Base" model (ViTt-B) introduced by [13], and also further configurations proposed in [15], namely the "Tiny" (ViT-T) and "Small" (ViT-S) models. In particular, ViT-T has 5×10^6 parameters organized in 12 layers, 3 MHA, $d = 192$, and adopts a one hidden layer MLP with 768 units; ViT-S has 22×10^6 parameters, 12 layers, 6 MHA, $d = 384$, and an MLP with 1536 neuron in the unique hidden layer; ViT-B has 86×10^6 parameters, 12 layers, 12 MHA, $d = 768$ and one hidden layer MLP with 3072 units. The images are rescaled to 224×224 to fit the input dimension of the vision transformer. The number of considered patches is $t = 196$, so the size of the single patch is 16×16. The experiments have been carried out on three datasets showing the method's reliability across multiple domains. The first dataset is the ImageNet dataset[16], the second is the FruitsGB dataset [17] and the third is the Cassava Dataset[18].

4.1. ImageNet

The ImageNet dataset[16] is the collection of data for the Large Scale Visual Recognition Challenge (ILSVRC), and it was proposed to evaluate algorithms for object detection and image classification. A rationale behind this dataset is to have a benchmark for researchers to compare progress for detecting a wide range of objects. Since it is largely users, it is used to measure the progress of large-scale image indexing in the tasks of retrieval and annotation[16]. All the model from Timm library[14] were pretrained with Imagenet dataset, for these experiments we added to the classic transformer architecture a further layer with 10 classes and fine tuned the network. A selection of images computed from the Imagenet dataset is shown in figure 2

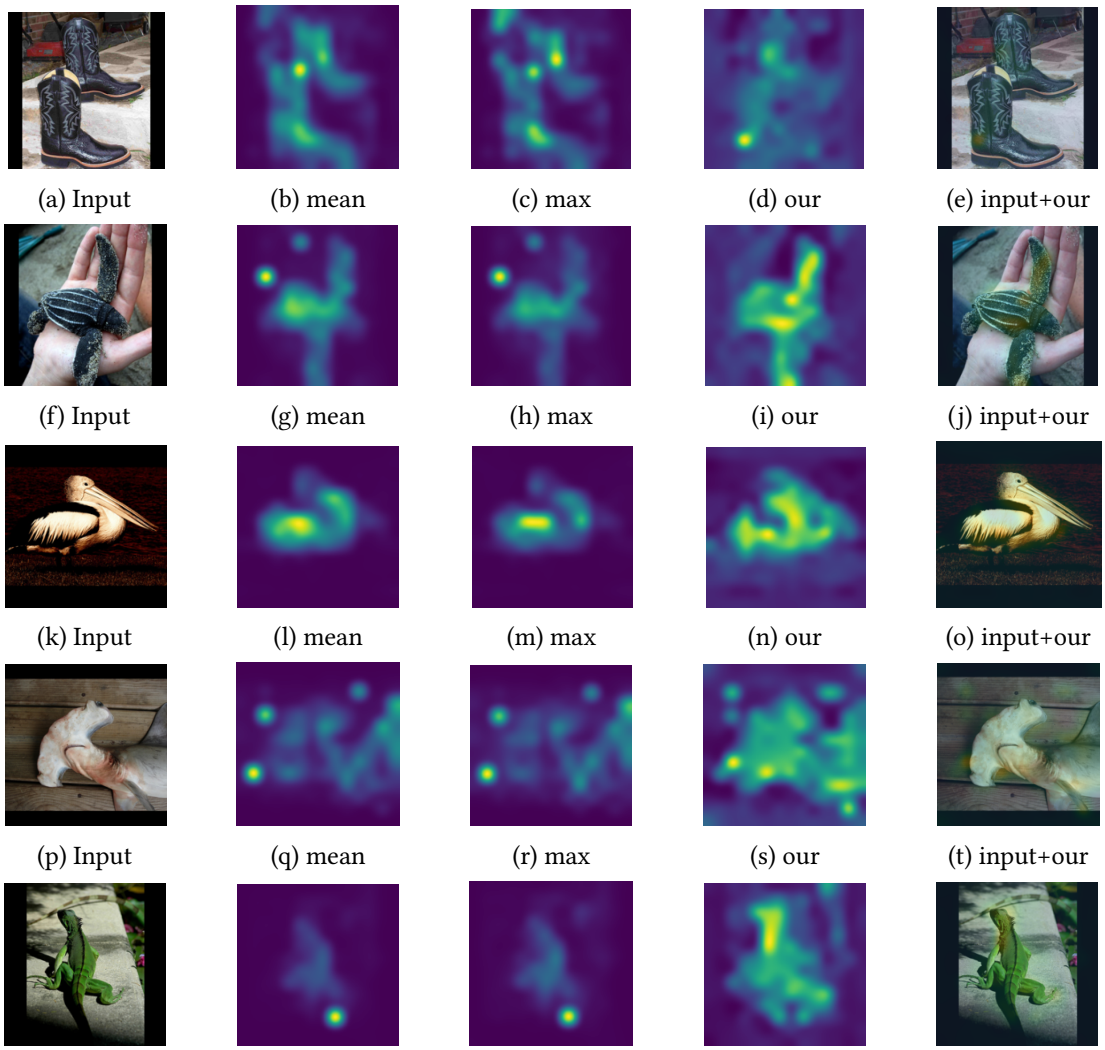


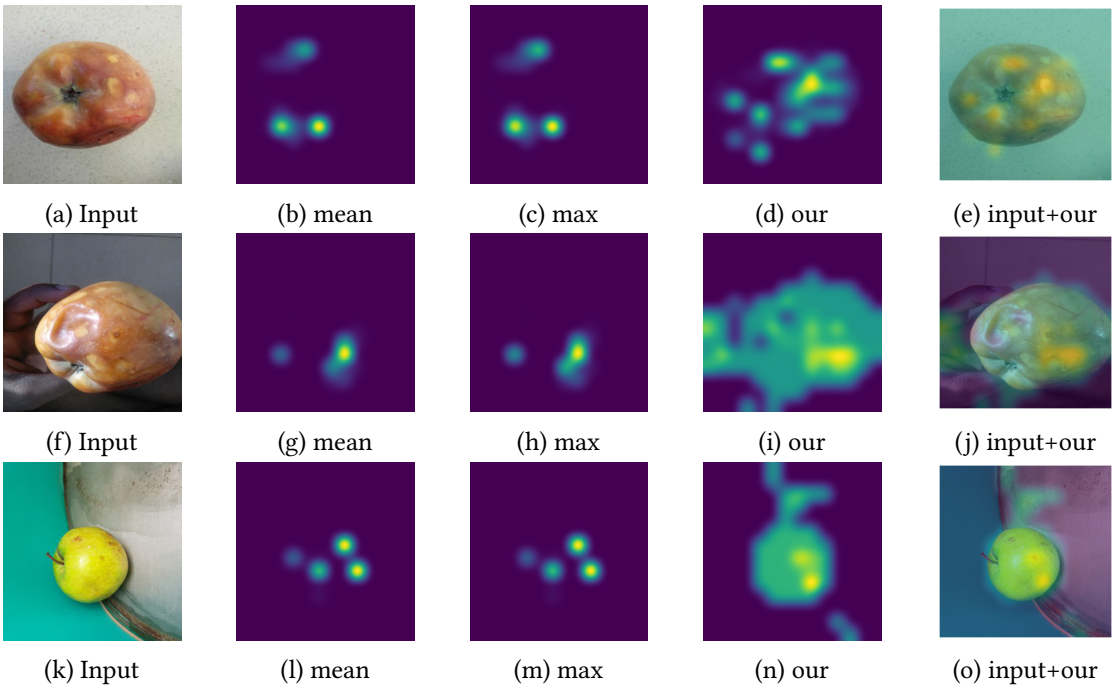
Figure 2: Example of attention activation for Imagenet dataset images through the composition of multi-head attention.

4.2. FruitsGB

The Fruits Good/Bad (FruitsGB) dataset[17] comprises 12000 images of 12 different classes of fruits: Bad Apple, Good Apple, Bad Banana, Good Banana, Bad Guava, Good Guava, Bad Lime, Good Lime, Bad Orange, Good Orange, Bad Pomegranate, and Good Pomegranate. The dataset is balanced: each class contains 1000 images. The images have a 256×256 resolution and were acquired using the mobile phone's rear camera with different angles, backgrounds, and lighting [17].

For the FruitsGB dataset, the ViT-tiny model, trained on the Imagenet dataset, was employed. The model has been fine-tuned on the samples of the Fruits dataset with training on 20 epochs.

The Adam optimisation has been chosen with a learning rate of 0.001 and a minibatch size of 32. The average accuracy is 0.96. Some example are shown in figure 3.



(a) Input (b) mean (c) max (d) our (e) input+our

(f) Input (g) mean (h) max (i) our (j) input+our

(k) Input (l) mean (m) max (n) our (o) input+our

Figure 3: Example of attention activation for FruitsGB dataset images through the composition of multi-head attention.

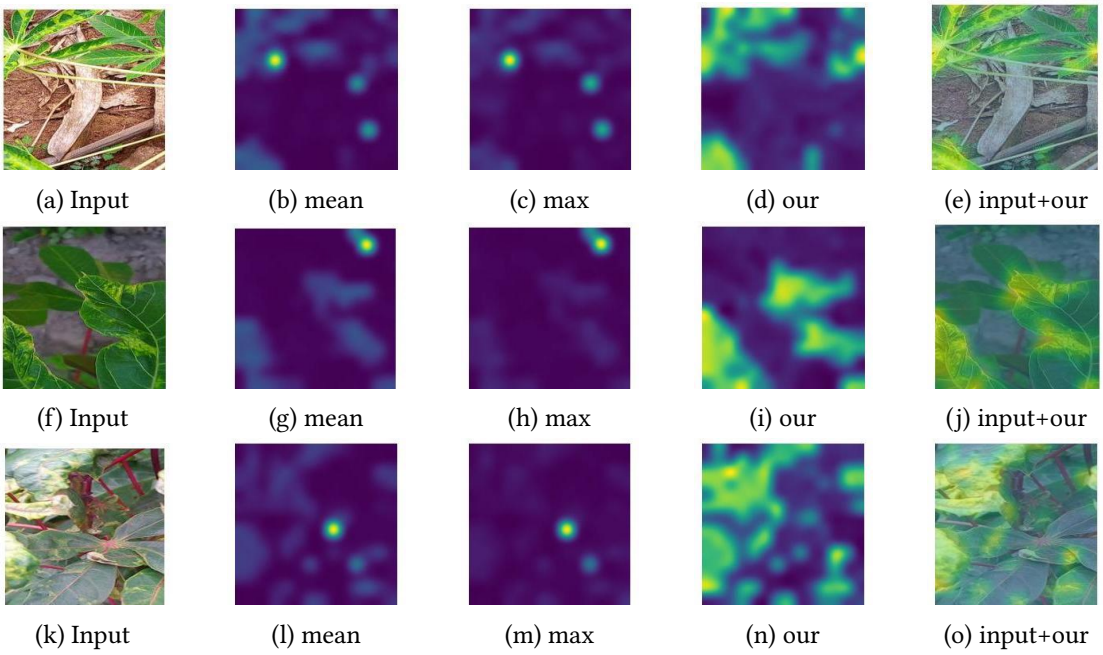
4.3. Cassava Dataset

The cassava is a food crop grown by small-holder farmers in Africa since it is a carbohydrates provider. It can be cultivate despite harsh conditions and it is an important source of food. These plants are affected by viral diseases that bring poor yields. The Cassava dataset[18] is composed of 9,436 cassava leaf images affected by four diseases that were annotated by experts at the National Crops Resources Research Institute (NaCRRI) in collaboration with the AI lab in Makarere University, Kampala. A fifth category for the healthy leaves has been added. The dataset was used for the fine-grained visual-categorization workshop (FGVC6) at CVPR 2019.

For the experiment with the Cassava Dataset the base transformer Timm model[14] was used. The training has been performed with a SGD optimizer with learning rate of 0.001 and momentum 0.9. The number of iteration has been set to 100. The final accuracy was 0.861. Some examples of the attention for the image of this dataset are shown in figure 4.

5. Conclusions

A method for visualizing attention in transformers has been shown and discussed. This technique shows the most relevant patterns for the deep model in the classification process, and it can be



(a) Input (b) mean (c) max (d) our (e) input+our

(f) Input (g) mean (h) max (i) our (j) input+our

(k) Input (l) mean (m) max (n) our (o) input+our

Figure 4: Example of attention activation for Cassava dataset images through the composition of multi-head attention.

useful for multiple purposes. The analysis of these areas allows an interested viewer to focus on these regions and inspect them with a deeper analysis. It is also a good starting point, even without knowledge of the specific domain, to assess if the found patterns are textured and in the central part of the image, or fall - as happen in some case - in homogenous areas along the images border. Multiple considerations about the relevant characteristics of the input samples and the proper training of the model can be drawn with this methodology.

6. Acknowledgments

Authors acknowledge the contribution of Giuseppe Marino in integrating the software module for attention and implementing the training and test procedures.

References

[1] W. James, The Principles of Psychology, Henry Holt and Company, 1890.
[2] L. Itti, C. Koch, Computational modelling of visual attention, Nature reviews neuroscience 2 (2001) 194–203.
[3] I. J. Goodfellow, Y. Bengio, A. Courville, Deep Learning, MIT Press, 2016.
[4] S. Cristina, What is attention?, 2022. URL: https://machinelearningmastery.com/what-is-attention/.

[5] G. W. Lindsay, Attention in psychology, neuroscience, and machine learning, Frontiers in computational neuroscience 14 (2020) 29.
[6] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, I. Polosukhin, Attention is all you need, Advances in neural information processing systems 30 (2017).
[7] S. Calderaro, G. L. Bosco, R. Rizzo, F. Vella, Deep metric learning for transparent classification of covid-19 x-ray images, in: 2022 16th International Conference on Signal-Image Technology & Internet-Based Systems (SITIS), IEEE, 2022, pp. 300–307.
[8] C. Molnar, Interpretable machine learning, Lulu. com, 2020.
[9] D. Amato, S. Calderaro, G. Lo Bosco, R. Rizzo, F. Vella, Metric learning in histopathological image classification: Opening the black box, Sensors 23 (2023). URL: https://www.mdpi.com/1424-8220/23/13/6003. doi:10.3390/s23136003.
[10] S. Calderaro, G. Lo Bosco, R. Rizzo, F. Vella, Deep metric learning for histopathological image classification, 2022, p. 57 – 64.
[11] H. Chefer, S. Gur, L. Wolf, Transformer interpretability beyond attention visualization, in: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, 2021, pp. 782–791.
[12] S. Abnar, W. Zuidema, Quantifying attention flow in transformers, arXiv preprint arXiv:2005.00928 (2020).
[13] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, N. Houlsby, An image is worth 16x16 words: Transformers for image recognition at scale, in: 9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021, 2021. arXiv:2010.11929.
[14] R. Wightman, Pytorch image models, https://github.com/rwightman/pytorch-image-models, 2019. doi:10.5281/zenodo.4414861.
[15] H. Touvron, M. Cord, M. Douze, F. Massa, A. Sablayrolles, H. Jégou, Training data-efficient image transformers & distillation through attention, in: International conference on machine learning, PMLR, 2021, pp. 10347–10357.
[16] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, L. Fei-Fei, ImageNet Large Scale Visual Recognition Challenge, International Journal of Computer Vision (IJCV) 115 (2015) 211–252. doi:10.1007/s11263-015-0816-y.
[17] V. Meshram, K. Thanomliang, S. Ruangkan, P. Chumchu, K. Patil, Fruitsgb: Top indian fruits with quality, 2020. URL: https://dx.doi.org/10.21227/gzkn-f379. doi:10.21227/gzkn-f379.
[18] T. G. ErnestMwebaze, Cassava disease classification, 2019. URL: https://kaggle.com/competitions/cassava-disease.