POLYPUBLIE

Improved Alzheimer disease diagnosis with a machine learning approach and neuroimaging: Case study development. Supplément

Lilia Lazli

Document en libre accès dans PolyPublie
Open Access document in PolyPublie

Date	2025
Type	Article de revue/Article
Version	Matériel supplémentaire / Supplementary material
URL de PolyPublie PolyPublie URL	https://publications.polymtl.ca/64723/
Évaluation par les pairs Peer reviewed	Révisé par les pairs / Refereed
Conditions d'utilisation Terms of use	Creative Commons Attribution 4.0 International (CC BY)
Référence Citation	Lazli, L. (2025). Improved Alzheimer disease diagnosis with a machine learning approach and neuroimaging: Case study development. JMIRx Med, 6, e60866 (13 pages). https://doi.org/10.2196/60866

Publication officielle
Official publication

Titre de la revue Journal title	JMIRx Med (vol. 6)
Maison d'édition Publisher	JMIR Publications Inc.
URL officielle Official URL	https://doi.org/10.2196/60866
Mention légale Legal notice	This is an open-access article distributed under the terms of the Creative Commons Attribution License (https://creativecommons.org/licenses/by/4.0/), which permits unrestricted use, distribution, and reproduction in any medium, provided the original work, first published in JMIRx Med, is properly cited. The complete bibliographic information, a link to the original publication on https://med.jmirx.org/, as well as this copyright and license information must be included.

Ce fichier a été téléchargé à partir de PolyPublie, le dépôt institutionnel de Polytechnique Montréal
This file has been downloaded from PolyPublie, the institutional repository of Polytechnique Montréal
https://publications.polymtl.ca

Multimedia Appendix 4: ViT

The ViT computation process with mathematical foundations is explained as follows:

We divided each input image (of dimensions $H \times W \times C$) to a grid (of dimension $(H/P) \times (W/P)$) containing a set of non-overlapped patches having a fixed size. Each of these patches functions as a token and contains a fixed number of pixels (16×16 pixels). Where H, is the height, W is the width, and C is the number of channels.

After that, the patches are transformed into a 1D embedding vectors sequence. After flattening, the image transforms to a token sequence, $x \in \mathbb{R}^{H \times W \times C}$. The image is flattened into a sequence of 2D patches $x_p \in \mathbb{R}^{CP^2 \times N}$. Where ($H$, W) is the image size, (P, P) is the size of each image patch, and $N = HW/P^2$ is the number of patches.

The flattened patches are then linearly projected onto lower dimensional vectors (token embeddings). Token embeddings allow to save patch content. So that each patch is vectorized and linearly projected into tokens by:

$$\hat{X} = [x_{class}, x_1\mathrm{E}, x_2\mathrm{E}, \dots, x_N\mathrm{E}], \mathrm{E} \in \mathbb{R}^{CP^2 \times D} \quad (4.1)$$

To provide spatial information, the token embeddings are augmented with learnable positional embeddings. These embeddings encode each token's relative position in the sequence. A positional embedding known as E_{pos} is added to the token as follows:

Token embeddings are complemented by learnable positional embeddings E_{pos} to supply spatial information. These latter integrations make it possible to encode the relative position of each token in the sequence.

$$\mathrm{Z}_0 = \hat{X} + \mathrm{E}_{pos}, \mathrm{E}_{pos} \in \mathbb{R}^{(N+1) \times D} \quad (4.2)$$

The vectors are then passed via a series of transformer blocks. Each transformer block has two sub-layers: a multi-head self-attention (MSA) layer and a feed-forward layer (MLP) with Layer-Norm (LN).

Next, the vectors flow through a series of transformer blocks that each contains a feed-forward layer (MLP) with layer-norm (LN) and a multi-head self-attention (MSA) layer.

At the end, at the output of the transformer blocks, we apply a classification head represented by a linear layer generating the probabilities of the input image class.