Using Image Processing and Computer Vision to Classify Blood Clot Origins of Ischemic Strokes

Antoniu Vadan[1], Mark Eramian[1]

[1]Department of Computer Science, University of Saskatchewan, Saskatoon, SK, Canada

Abstract

Ischemic strokes are caused by interrupted or reduced blood supply to the brain due to blood clot formations in arteries which prevent the brain from getting the oxygen and nutrients it needs [2]. Over 700,000 people experience an ischemic stroke in the United States every year [1]. This project aimed to develop and evaluate a deep learning model to classify histopathological slides of two blood clot classes based on stroke etiology subtypes: cardioembolism (CE) and large artery atherosclerosis (LAA). The dataset used in this project is publicly available as part of a Kaggle competition. The photographs that constitute the dataset are of very high resolution. Our approach is to decompose each high-resolution image into square tiles of equal size, use an Inception-V3 [5] variant to classify the tiles, and use a simple voting mechanism to classify the image that the tiles compose. A total of 754 images are included in the dataset, of which 72.5% are CE and 27.5% are LAA blood clots. 15% of images were reserved for the test set without the introduction of bias via undersampling or oversampling. The classification accuracy of our model is 64.4% for tiles and 70.4% for images. While the performance of our model is not optimal, our experiment shows that the voting mechanism implemented is a useful tool for classifying high-resolution images.

1. Introduction

Two major acute ischemic stroke etiology subtypes are cardioembolism and large artery atherosclerosis [1]. The goal of this project is to create a system that classifies high resolution histopathology images of blood clots into the two etiology subtypes listed above. Knowledge of ischemic stroke etiology helps physicians prescribe optimal post-stroke therapy to patients which reduces the likelihood of subsequent strokes [1].

The application of artificial intelligence to medical diagnostic disciplines (such as radiology, pathology) has shown excellent results [9]. Diagnostic disciplines rely heavily on pattern recognition in data by physicians and the interpretation of such patterns [8]. However, reproducibility among physicians has shown to be suboptimal [8, 10, 11]. The automation of pattern recognition in medical data, when accurate, can improve diagnostics by making them more efficient and reproducible [8].

We leverage deep neural networks to classify images of blood clots as one of the two stroke etiology subtypes. As the dataset is composed of high-resolution images, they cannot be used as input for neural networks without prior downsampling. However, downsampling the images to match the input dimensionality requirements of most neural networks requires a high downsampling factor (in our case, as high as 150). Therefore, our approach is to decompose each image into equally sized tiles and downsample the tiles instead. This approach results in the use of a much smaller downsampling factor, thus preserving more of the information from the original images. A deep neural network is trained to predict the class that a tile belongs to. In addition to passing the tiles as input to the neural network, we separately extract two image feature descriptors, one from the tile and one from the image the tile is part of, and pass them as input to the neural network as well. To classify the images in the dataset we use a simple voting mechanism by which an image is classified as the majority predicted class of its constituent tiles. Section 3 is dedicated towards describing the dataset, the techniques used to preprocess the images in the dataset, the features extracted from the tiles and images and, finally, the neural network architecture used to classify the tiles.

2. Related works

Convolutional neural networks (CNNs) and recurrent neural networks (RNNs) have demonstrated great performance in histopathological classification tasks. Iizuka et al. have produced models achieving areas under the receiver operating characteristic curve (AUCs) of ≥ 0.96 in classifying histopathological images of gastric and colonic epithelial tumours [12]. Byeon et al. have also shown excellent results in discriminating adenocarcinoma from non-

adenocarcinoma lesions using CNNs, with AUCs ≥ 0.995. Additionally, they have grouped histopathological slides of colonoscopic biopsy or resection specimens into six classes by disease category. The mean classification accuracy of their top performing model was 97.3% [13]. Bejnordi et al. compared the performance of deep learning algorithms at detecting lymph node metastases in women with breast cancer with the performance of pathologists' diagnoses. The AUCs for their models ranged from 0.556 to 0.994. Their top-performing model achieved a true-positive rate comparable with that of the pathologists'. For the whole-slide classification task, the best model performed significantly better than the pathologists (AUC of 0.994 versus 0.810) [14]. Ertosun et al. classified images of brain glioma of different pathologic subtypes using convolutional neural networks. They observed a classification accuracy of 96% for the task of distinguishing between two primary pathologic subtypes. Their top performing model had a sensitivity of 0.98 and specificity of 0.94 [19].

2. Methods

2.1. Data

The dataset used in this project was made publicly available by a Kaggle competition authored by the Mayo Clinic, an American medical research institution. The dataset consists of a total of 754 images. The blood clots were extracted via mechanical thrombectomy. 72.5% of images are of CE blood clots, and 27.5% are LAA blood clots. 15% of the images were reserved for the test set, whose class distribution closely matches that of the entire dataset. The dataset consists of high-resolution histopathology images of blood clots. Dimensions vary significantly across images, some images being as large as 73777×38568 pixels.

2.2. Image preprocessing

To aid in the downstream separation of foreground pixels from background pixels, images were denoised by performing a grayscale conversion and setting the intensity of pixels with original intensity less than 0.8 to 1 (white). All white pixels in the grayscale image were set to white in the original RGB version of the image. Subsequently, images were divided into 4096×4096-pixel tiles, each downsampled by a factor of 16 to create 256×256 tiles. Tiles with 80% or more background pixels were discarded. Figure 1 shows an image's tile decomposition.

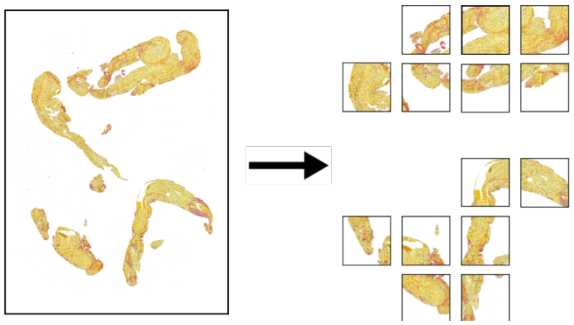

Figure 1: Tile decomposition. Only valid tiles are shown.

One of the two feature extraction algorithms used in this project accepts binary images as input. Separating the foreground pixels from the background pixels was a simple process thanks to the denoising mentioned previously. We applied three operations on the binary images to further remove noise. We applied closing with a disk of radius 4 to smoothen the boundary of the foreground region. We did so as we wanted to preserve the general shape of the foreground region while discarding granular variations which do not affect the high-level shape. The closing was followed by an opening, which removes tiny foreground regions from images. Finally, we filled each hole in the foreground region as the feature extraction algorithm that accepts binary images is strictly concerned with the outer boundary of the foreground region. Figure 2 shows an example of an image's resulting binary counterpart.

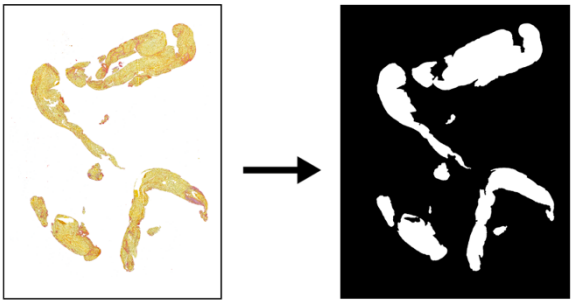

Figure 2: The result of separating the foreground pixels from the background pixels in an RGB image and applying a closing, opening, and filling.

2.3. Feature extraction

Mardanisamani et al. combined handcrafted features and CNN features to classify images of crops. The use of handcrafted features can increase model accuracy, particularly for tasks where only a limited amount of training data is available [3]. We extracted the histograms of curvature (HoC) shape descriptor from the whole images to capture the global shape of each blood clot and the local binary patterns (LBP) texture

descriptor from the tiles. Histograms of curvature encode shape by computing the local curvature for each point along the boundary of the foreground region. Local curvature is calculated as the ratio of pixels in a boundary point's neighbourhood that lie within the foreground over the total area of the neighbourhood (see figure 3) [15]. The algorithm computes local curvature for each point along the foreground's boundary and places the calculated proportion of neighbouring pixels that lie within the foreground region in one of many bins which, when aggregated, produce a histogram that encodes the foreground region's shape. Histograms of curvature were computed for three radii values of the circular neighborhood used to compute local curvature: 5, 15, and 25 pixels.

At a high level, the LBP texture descriptor encodes local texture by calculating the number of neighbouring pixels whose grayscale intensity is greater than that of the current pixel (see figure 3). Performing this computation for each pixel in the image and placing each result in one of many bins yields a histogram that encodes a tile's texture. The LBP texture descriptor was created using 8- and 24-bit (i.e. neighborhoods of size 8 and 24 pixels respectively) rotationally invariant uniform local binary patterns with neighborhood radius of 1 and 3 pixels respectively. These LBP parameters have produced the best classification accuracy in the texture classification experiments of Ojala et al. [4]. Feature values were normalized to a mean of 0 and standard deviation of 1.

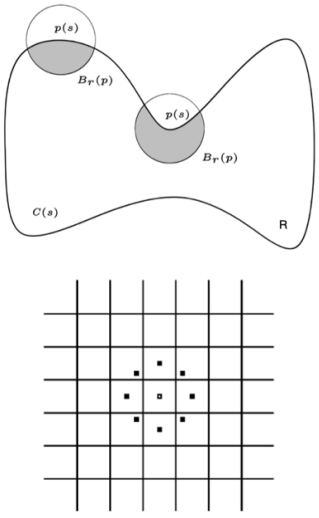

Figure 3: Top: local curvature is computed as the proportion of pixels in the neighborhood of a boundary pixel that lie in the foreground. Image credit: Manay et al. [15]. Bottom: LBPs count the number of neighboring pixels whose intensity is greater than that of the current pixel. Image credit: Ojala et al. [4].

2.4. Neural network architecture

The neural network architecture is inspired by the architecture proposed by Mardanisamani et al. [3]. It consists of three high-level modules: the CNN backbone, a feature extraction module, and a classification module. Our CNN backbone consists of Inception-V3 with ImageNet weights preloaded [5]. Our choice of Inception-V3 was motivated by its performance reported in an analysis of deep neural network model architectures by Canziani et al. [6] (see figure 4) as well as its accessibility as a PyTorch module. In this analysis, Inception-V3 ranked second in classification accuracy in the ImageNet challenge. Inception-V4 was the top performer in the ImageNet challenge according to the survey conducted by Canziani et al. but, due to its unavailability as a PyTorch module, we decided to use its predecessor as our network's CNN backbone.

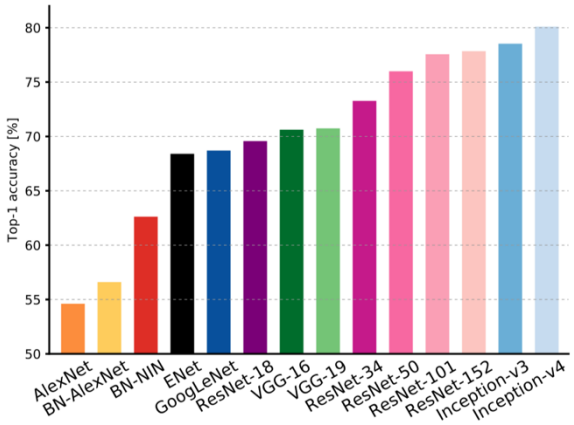

Figure 4: Top-1 classification accuracy of the most relevant entries submitted to the ImageNet challenge. Inception-V3 is second only to Inception-V4, which, at the time of this writing, is not available as a PyTorch module. Image credit: Canziani et al. [6].

The feature extraction module extracts the features described in section 2.4 ahead of time. It concatenates the resulting feature descriptors into a 1-dimensional vector. The HoC shape descriptor is extracted from the image that the input tile is part of, such that global shape information is preserved, while the LBP texture descriptor is extracted from the input tile. The handcrafted feature vector is concatenated with the 1-dimensional output of the CNN backbone and the result is passed to the classification module, which consists of the result of the previous concatenation, a 128×1 layer, and a final 2×1 layer. Figure 5 shows the architecture proposed by Mardanisamani et al. [3], which guided our implementation.

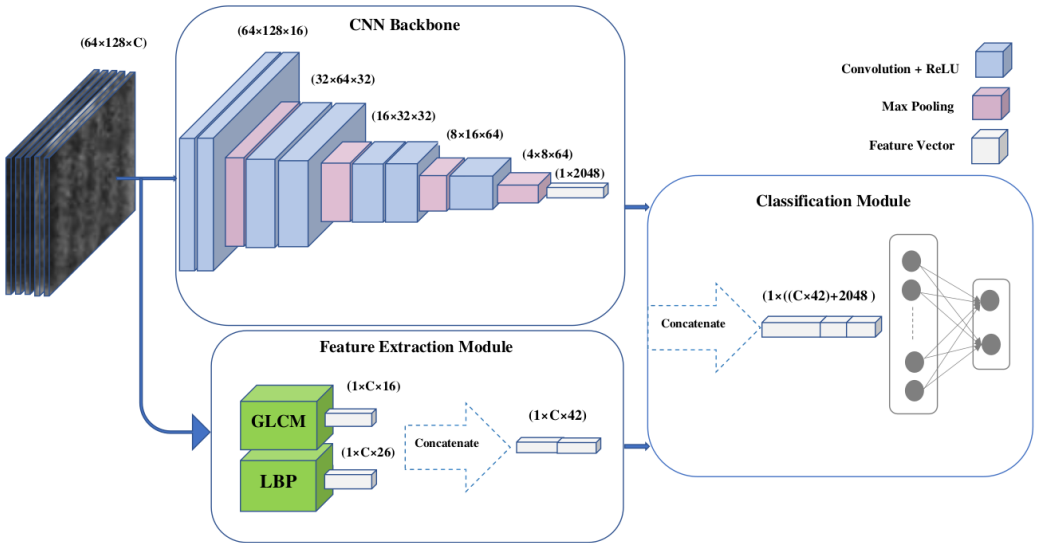

Figure 5: Our network architecture consists of three high-level modules: CNN backbone, feature extraction module, and classification module. Notable differences between LodgedNet [3] and our DNN is the use of Inception-V3 as the CNN backbone and the extraction of HoCs and LBPs in the feature extraction module. Image credit: Mardanisamani et al. [3].

3. Results

We trained three machine learning models on the handcrafted feature data, where both feature descriptors were extracted from whole images (as opposed to extracting LBPs from tiles). Two support vector machine (SVM) variations, the former with a linear kernel (C = 0.001) and the latter with a radial basis function kernel (C = 10 and gamma = 0.001), and a random forest classifier (with four maximum leaf nodes and 100 estimators) were trained.

Model	Classification accuracy (%)
SVM – linear kernel	68.31
SVM – radial basis function kernel	72.81
Random forest	72.28

Table 1: Classification accuracies of three machine learning models trained on handcrafted feature data. Classification accuracy obtained via 5-fold cross-validation.

Model hyperparameters were selected via grid search. The hyperparameter search space for C for the SVM with linear kernel was 0.001, 0.01, and 0.1. The search space for the C hyperparameter for the SVM with radial basis function kernel was 0.01, 0.1, 1, 10, and 100, and the search space for the gamma hyperparameter was 0.0001, 0.001, 0.01, 0.1, 1, and 5. The random forest classifier's leaf nodes hyperparameter search space consisted of 4, 8, 16, and 32, and the number of estimators hyperparameter search space consisted of 10, 100, 250, and 500. 5-fold cross-validation performance is shown in table 1. Our choice to train and evaluate the performance of a set of machine learning models was motivated by wanting to establish a baseline classification accuracy to compare our more complex DNN-based classifier's performance against.

The network was trained to classify tiles over 20 epochs with a batch size of 16. Figure 6 shows the training accuracy and the validation accuracy. The highest training and validation accuracies across all epochs are 71.4% and 72.5% respectively. The tile test accuracy is 64.4% while the image test accuracy, obtained via the voting mechanism described in section 2.1., is 70.4%.

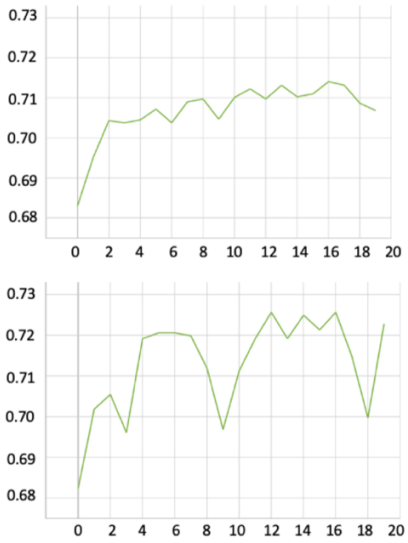

Figure 6: Neural network training accuracy (top) and validation accuracy (bottom) over 20 epochs.

4. Discussion

The 7% difference between tile training accuracy and tile test accuracy is a sign that the network is overfitting. Typical causes of overfitting include the training set being too noisy, not having enough data, and the model being too complex [7]. Having taken steps to reduce noise in the data, the second and third causes of overfitting are more likely in this scenario. Possible solutions to overfitting are using a simpler CNN backbone, increasing dropout through the network, and using data augmentation methods.

The image test accuracy of 70.4%, being less than the proportion of CE images in the test set (72.5%), suggests that the image characteristics that distinguish CE and LAA blood clots are not evenly distributed throughout the images. However, the image test accuracy (70.4%) being greater than the tile test accuracy (64.4%) suggests that there is merit in tiling images and using a voting mechanism to produce classification results for images. However, using a different weighting of the tile votes, as opposed to the equal weighting used in our experiment, may lead to better results.

5. Future work

To get a complete representation of our model's performance, the sensitivity and specificity must be computed. Computing the AUC for our model would allow us to compare its performance to other classifiers of histopathological data [12, 13, 14, 19]. Additionally, oversampling LAA blood clots and undersampling CE blood clots in our dataset to generate a test set with an even distribution of blood clot types is a superior way of measuring the performance of our model.

The neural network architecture used in this project was primarily motivated by the results shown by Mardanisamani et al. [3]. We did not, however, compare the performance of our network to a variation of it with the feature extraction module removed. Measuring the difference in performance can help researchers and engineers make more informed decisions in neural network architecture: if the improvement is negligible, the feature extraction module may be removed entirely for resource constrained applications.

To address the observed overfitting of our model, simpler CNN backbones may be used in future work. Drawing inspiration from Byeon et al. [13], EfficientNet-B7 and DenseNet-161 seem good candidates.

The hypothesis that a different tile weighting would perform better can be tested by using transformer models to classify the blood clots, as transformers [16, 17] learn the significance of different parts of the input data. Transformers have demonstrated great performance in image classification tasks [18].

6. Acknowledgements

We would like to thank MSc. student Tim Wheler for assisting in setting up the compute infrastructure to train our neural network on the University of Saskatchewan's GPU servers. We would like to thank Dr. Jordan Ubbens for providing guidance regarding the implementation of handcrafted feature injection in the neural network. We would like to thank PhD Student Sara Mardanisamani for allowing us to refer to her implementation of LodgedNet [3].

7. References

[1] "Mayo Clinic - STRIP AI." Kaggle, 2022, https://www.kaggle.com/competitions/mayo-clinic-strip-ai.

[2] "Stroke - Symptoms and causes." Mayo Clinic, 20 January 2022, https://www.mayoclinic.org/diseases-conditions/stroke/symptoms-causes/syc-20350113.

[3] Mardanisamani, Sara, et al. "Crop Lodging Prediction from UAV-Acquired Images of Wheat and Canola using a DCNN Augmented with Handcrafted Texture Features." 2019.

[4] Ojala, Timo, et al. "Multiresolution Gray Scale and Rotation Invariant Texture Classification with Local Binary Patterns." 2002.

[5] Szegedy, Christian, et al. "Rethinking the Inception Architecture for Computer Vision." 2015.

[6] Canziani, Alfredo, et al. "An Analysis of Deep Neural Network Models for Practical Applications." 2017.

[7] Géron, Aurélien. "Hands-on Machine Learning with Scikit-Learn, Keras, and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems." O'Reilly, 2019.

[8] van der Laak, Jeroen, et al. "Deep learning in histopathology: the path to the clinic." Nature Medicine, 2021.

[9] Yu, KH., Beam, A.L. & Kohane, I.S. "Artificial intelligence in healthcare." Nat Biomed Eng 2, 719–731 (2018). https://doi.org/10.1038/s41551-018-0305-z

[10] Bulten, W. et al. "Automated deep-learning system for Gleason grading of prostate cancer using biopsies: a diagnostic study." Lancet Oncol. 21, 233–241 (2020).

[11] Ehteshami Bejnordi, B. et al. “Diagnostic assessment of deep learning algorithms for detection of lymph node metastases in women with breast cancer.” JAMA 318, 2199–2210 (2017).
[12] Iizuka, O., Kanavati, F., Kato, K. et al. “Deep Learning Models for Histopathological Classification of Gastric and Colonic Epithelial Tumours.” Sci Rep 10, 1504 (2020). https://doi.org/10.1038/s41598-020-58467-9.
[13] Byeon, Sj., Park, J., Cho, Y.A. et al. “Automated histological classification for digital pathology images of colonoscopy specimen via deep learning.” Sci Rep 12, 12804 (2022). https://doi.org/10.1038/s41598-022-16885-x.
[14] Ehteshami Bejnordi B, Veta M, Johannes van Diest P, et al. “Diagnostic Assessment of Deep Learning Algorithms for Detection of Lymph Node Metastases in Women With Breast Cancer”. JAMA. 2017;318(22):2199–2210.
doi:10.1001/jama.2017.14585
[15] Manay, S., Cremers, D., Hong, B., Yezzi, A., Soatto, S. “Integral Invariants for Shape Matching”. 2006.
[16] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A., Kaiser, Ł. “Attention Is All You Need.” Dec, 2017.
[17] Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., Uszkoreit, J., Houlsby, N. “An Image is Worth 16x16 Words”. Jun, 2021.
[18] Khan, S., Naseer, M., Hayat, M., Zamir, S., Khan, F., Shah, M. “Transformers in Vision: A Survey.” Jan, 2022.
[19] Ertosun, M., Rubin, D. “Automated Grading of Gliomas using Deep Learning in Digital Pathology Images: A modular approach with ensemble of convolutional neural networks.” 2015.