Multidisciplinary Journal of Research in Engineering and Technology

ISSN: 2348-6953

Volume 13 Issue 2s (2026) | Pages: 246-254

HemorrhageNet: Automated Detection and Fine-Grained Subtype Identification of Intracranial Hemorrhage in Non-Contrast CT Imaging Using Calibrated Dual-Head EfficientNetB4

Harshwardhan Ahire[1], Sakshi Bhosale[2], Isha Gandecha[3], Pranav Kadam[4]

[1,2,3,4]Department of Electronics and Telecommunication Engineering (EXTC), K J Somaiya Institute of Technology, Mumbai, India
[1]harshwardhan@somaiya.edu;[2] sakshi.db@somaiya.edu; [3]isha.gandecha@somaiya.edu; [4]pranav.vk@somaiya.edu

Peer Review Information

***Type**: Article*
***Received**: 27 March 2026*
***Revised**: 12 April 2026*
***Accepted**: 26 May 2026*
***Published**: 16 June 2026*

Abstract

Finding out if someone has a brain bleed, which is also called a hemorrhage is very important. We use a kind of x-ray called a cranial computed tomography or CT scan to check for this. If we do not find out about the brain bleed it can cause very bad damage to the brain or even death. In some places there are no doctors who are experts in looking at these x-ray pictures and they can get very tired from working too much. This means that sometimes people do not get the care they need . We can use computers to help look at these pictures and find out if someone has a brain bleed.

This paper is about a computer program called Hemorrha- geNet. It is a kind of program that uses pictures to decide if someone has a brain bleed and what kind of bleed it is. There are five kinds of brain bleeds: Intraventricular, Intraparenchymal, Subarachnoid, Epidural and Subdural. The program uses a kind of picture called a CT slice and it was trained on 1,200 of these pictures.

The program is very good at finding out if someone has a brain bleed. It is right 93.9 percent of the time. It never misses a bleed. The program is also good at figuring out what kind of bleed it is. It is about 80 percent of the time for each of the five kinds of bleeds.

We think that HemorrhageNet can be very helpful in hospitals. It can help doctors find out if someone has a brain bleed and what kind of bleed it is. This means that doctors can make decisions about how to take care of the person.The program uses a kind of computer model called a convolu- tional model. It was trained on 1,200 CT slices.It is very good at finding out if someone has a brain bleed and what kind of bleed it is. It can help doctors make decisions, about how to take care of people with brain bleeds.

Keywords: Intracranial Hemorrhage; EfficientNetB4; Focal Loss; Threshold Calibration; CT Image Classification; Computer-Aided Diagnosis.

How to Cite This Article

Ahire, H., Bhosale, S., Gandecha, I., & Kadam, P. (2026). HemorrhageNet: Automated detection and fine-grained subtype identification of intracranial hemorrhage in non-contrast CT imaging using calibrated dual-head EfficientNetB4. *Multidisciplinary Journal of Research in Engineering and Technology*, 13(2s), 246–254.

Introduction

Intracranial hemorrhage is not one condition it is a whole range of conditions where blood collects in or around the brain. These conditions make up one out of every eight stroke cases. The numbers are really bad: for some types of hemorrhage almost 40 percent of patients die soon after. Time is very important here. Doctors have to act fast. With a blocked artery you might have a hours to act but with intracranial hemorrhage doctors have to operate within minutes. There is no time to waste.

In emergency cases, CT scans are used by physicians using contrast because they are fairly quick and reliable to perform. The major benefit is that doctors can easily see blood in the brain with a CT scan. Reading a CT scan of the head is much more complex than interpreting images from other parts of the body and requires that the reader (doctor) examines many images to view subtle differences, such as blood vs. calci- fication, under intense conditions where there are significant time constraints imposed by the urgencies of the situation. Furthermore, due to the vast experience that most physicians have, they are still capable of making mistakes in reading CT scans. The advancement of machine learning (ML) as well as large, convolutional neural networks (CNNs) has afforded the opportunity to automate the reading of CT scans using much fewer examples of images than originally thought to exist. The EfficientNet models have demonstrated exceptional accuracy and low consumption of computer memory, both of which are important for simultaneous scan capability.

One problem is that public datasets are not balanced: most scans do not show bleeding and some types of bleeding are rare. Another problem is that the brains anatomy is complex: bleeding can happen in areas and patients can bleed into multiple regions at once. This can confuse models that are set up to make one decision per scan. Also missing a bleed can be fatal while calling a positive is usually just extra work. Most research focuses on "'s there blood or not?" without looking at specific types of bleeding even though this has a direct impact on what doctors do next. Managing types of intracranial hemorrhage is not the same.

This paper addresses these problems with the following ideas:

- We created a dataset protocol that is easy to reproduce. We took CT scan slices made the contrast better and added many different kinds of changes to the pictures. This gave us 200 examples for each class.
- We made a dual-head EfficientNetB4 model that is effi- cient with parameters. It can do two tasks at once: spot- ting hemorrhage and breaking it down into five subtypes. Both tasks share the features but have different outputs.
- We used a loss function for training and added label smoothing to avoid overconfident guesses.
- We also developed a method to pick the thresholds for each subtype. It tries different values and chooses the one that gets the highest score but only if the accuracy is, above 80
- As a result, we evaluated the performance of the model using 180 images and sent those images for analysis to determine both (a) where our model failed and (b) how and/or why that failure poses risk or concern in practice. One potential use of our model would be to help physicians make early diagnoses of intracranial hemorrhaging

Problem Analysis and Challenges

Epidemiological Imbalance in Clinical Cohorts

Emergency room records generally are not uniform across patients. You might see a lot of standard images on all patients and a few images with disease and almost no patients that had epidural dye and the other types will be more common. When you train a network on Clinical datasets from real emergency departments like this and do not do anything to address the imbalance the model mostly learns from the majority class, which is the normal scans. It gets good accuracy overall just by guessing that the scans are normal most of the time and it misses the rare cases, like Epidurals altogether.

The standard loss functions do not help with datasets from real emergency departments. If you use binary cross-entropy every mistake gets treated the same way so the model learns just as much from an easy normal case as it does from a hard-, to-catch Epidural case so it never really learns to spot those tricky rare subtypes, like Epidurals.

Co-occurrence and Multi-Label Formulation

Softmax output layers assume each case belongs to one class.That doesn't work for ICH subtyping.Traumatic brain injuries often cause bleeding in than one area at the same time.Someone who suffers a serious injury may have subdural and subarachnoid Hemorrhages at the same time. Classifying these cases using Argmax is not only misleading when looking at the subtypes of ICH, but it also limits the model's ability to learn about the subtypes of ICH. A multi-label sigmoid configuration is preferred for ICH subtyping because it allows the model to make independent predictions on each specific subtype of ICH, thus allowing the model to learn more about cases that have overlapping types of ICH. This will ensure that every ICH edge has equal importance and can provide a good representation of ICH from ICH subtype's perspective. ICH is challenging to classify into sub-types. In addition to being a challenging classification problem, ICH

will typically need a very sophisticated model to differentiate among its subtypes.

Inter-Subtype Visual Ambiguity

All kinds of brain hemorrhages appear brighter than normal brain tissue on a CT scan. The challenge comes because each type of hemorrhage tends to have its own pattern for a given type of subarachnoid hemorrhage, for example, fol- lows grooves in the brain and curves along the sulci and cisterns. Subdural bleeds press up against the skull, form- ing crescent-shaped areas on the scan. On the other hand, epidural hematomas appear as lens-shaped images on CT, are very bright, and are typically located between the skull and protective membrane or duramater. In real life, however, CT scans can be very messy. For example, if a lot of blood has crossed into different areas around the brain, it can create challenges for diagnosing. Additionally, many experienced radiologists will question what they are seeing when the bleed on the scan does not match their own understanding of the classic characteristics of that particular bleed type. The way radiologists diagnose a hemorrhage is by looking at slices on the CT.

Threshold Sensitivity and Clinical Cost Asymmetry

Sigmoid classifiers give us probabilities. The threshold we choose is what really matters when it comes to how our model works on a precision-recall curve. We might think that setting one threshold for everything to get the overall accuracy is a good idea.. This can actually cause problems. Some types of things the serious ones can have a lot of false negatives. This means we miss them when we should not. Other types can have a lot of positives. This means we think they are something when they're not.Every type of thing has its way of giving scores. This is because of how common it's how easy it is to find and how we changed the data. So using one threshold for everything does not work well.What works better is to use a set of data to calibrate our model. We make sure each type of thing is found with a level of accuracy. This way we can trust our model more. Sigmoid classifiers and their thresholds are important. We have to think about how we use them.

Proposed Methodology

Source Data and Selection Criteria

We did our experiments using the PhysioNet Intracranial Hemorrhage Detection and Segmentation dataset version 1.0.0. The PhysioNet Intracranial Hemorrhage Detection and Seg- mentation dataset is a set of -contrast cranial CT scans that we can get for free. These scans have a lot of details in the annotations.Each annotation in the PhysioNet Intracranial Hemorrhage Detection and Segmentation dataset has an ID and a slice index. The annotation also tells us about five hemorrhage subtypes. It even tells us if there is no hemorrhage all.If we did not have a brain-window file for a slice we used the bone-window image from the PhysioNet Intracranial Hemorrhage Detection and Segmentation dataset instead.. If we could not find either the brain-window file or the bone- window image, for a slice we just left that slice out. We kept track of how slices we had to skip in the PhysioNet Intracranial Hemorrhage Detection and Segmentation dataset.

Preprocessing: CLAHE Enhancement

First used OpenCV to decode the CT JPEG images. Then I made the images smaller to 224×224 pixels using a way of making pictures smaller called bilinear interpolation. After that each image went through something called Contrast- Limited Adaptive Histogram Equalisation or CLAHE for short. CLAHE works by splitting the image into squares like an 8×8 grid so each square gets its own special way of making the image look better. I did this so that the image would look even. I also did not want to make the noise in the image worse in the flat areas like the cerebrospinal fluid spaces so I set a limit to stop this from happening the limit was 2.0. This helps to keep the noise under control. When you do this the boundaries around the hemorrhages in the brain become really clear and easy to see they stand out more from the brain tissue around them. The image becomes clearer. It is easier to find the important things in the image. At the time the brightest spots, in the image do not get too bright because CLAHE stops them from getting too saturated. This makes the image look better. It is easier to work with the CT JPEG images.

Corpus Balancing through Controlled Augmentation

Getting labeled data is a big problem in medical AI. This is especially true when some types of images are more common than others.To solve this we use a step-by-step process to increase the number of images. We aim for 200 images per type.First we collect many real images as possible for each type. If we don't have 200 we create images to fill the gap.We use methods to create these fake images. These methods include:

- Flipping the image sideways. This works because the brain is symmetrical.
- Flipping the image up and down. This works for parts.
- Rotating the image a bit up to 15 degrees.
- Increase or decrease brightness of image to an extent.
- Slightly adjust image contrast.

- Add noise to image. Noise simulates an image from a quality CT scan.
- Combine image rotation and brightness.
- Blurred images create the appearance of a quality CT scan in images with slices.

All these methods will continue to be combined in different combinations and shuffled together with a maximum of 200 images in total per type.

Dual-Head Architecture

The main part of this system is EfficientNetB4, a network. It gets bigger in three ways: depth, width and the size of the input. All of this is based on some numbers that have been tested and work well. The network uses weights that it learned from ImageNet, which helps it recognize patterns in images right away.The model also has a step that gets the image ready. It takes the pixel values which're between 0 and 1 and changes them to the BGR range that EfficientNetB4 expects. This way when you use the model later you will not have any problems with the inputs not matching.

At the center of the network the spatial feature maps from the part of EfficientNetB4 go through two paths at the same time. One path is average pooling, which finds the average activation across the whole image. The other path is max pooling, which finds the highest activation in each channel, which is great for finding small very bright lesions. Each path creates a vector with 1792 dimensions. When you put these vectors together you get a mix with 3584 dimensions that combines both types of information.

At the top the model splits into two parts. One part is a sigmoid neuron that predicts whether there is a hemorrhage or not. The other part is a five-unit sigmoid layer, where each output gives the probability for an anatomical subtype of EfficientNetB4. Since each subtype output is independent the model can handle cases, with labels without any problems.

Asymmetric Focal Loss with Label Smoothing

The output heads use a loss objective. When the model makes a prediction it looks at how sure it's about that prediction. The focal loss helps the model pay attention to the things it is already sure about.

The focal loss is calculated using a formula:

$$L = - \cdot (1 - p) \cdot [\, y \cdot \log(p) + (1-y) \cdot \log(1-p) \,]$$

Here p is how sure the model is about what's really true. The helps the model not get too confused, between things that're are not true.For one part of the model we use = 2.0 and = 0.75. For another part of the model we use = 2.0 and = 0.70.For the part of the model we also do something called label smoothing. This means we do not say something is absolutely true or false.Instead we say it is probably true or probably false. We do this because the model can get a little mixed up when it looks at things that're not perfect examples.This helps the model not get too extreme in its predictions.

Two-Phase Training Strategy

Phase 1. Controlled Warm-Up (15 epochs): We start by keeping the EfficientNetB4 backbone frozen. Only the new dense tower and output heads get updated. This helps prevent gradients from new heads damaging the pre-trained feature detectors. We use the Adam optimiser with a learning rate of 0.0003.

Phase 2. Selective Backbone Fine-Tuning (up to 60 epochs): Next we unfreeze the 60 percent of backbone layers and update them together with the dense heads. The lower 40 percent stay fixed to keep the edge and texture detectors. We lower the learning rate to 0.00003 to avoid messing up the backbone. We have three rules for training:

- ModelCheckpoint saves the model based on validation binary AUC.
- ReduceLROnPlateau cuts the learning rate in half after four epochs with no improvement in validation AUC.
- EarlyStopping stops training. Goes back, to the best model if theres no improvement after 14 epochs.

Validation-Set Threshold Calibration

After we finish training we use the model to make predic- tions on the validation data. We do this for each of the five subtypes. For each subtype we try out thresholds to see which one works best. The thresholds we try are 0.30, 0.32 and so on, up to 0.84. For each threshold we make predictions. Calculate how accurate they are and what the F1-score is. We choose the threshold that gives us the F1-score as long as the accuracy is at least 0.80. If none of the thresholds work enough we use

0.55 as the default. The threshold, for the head always stays at 0.55.

Six-Class Inference Fusion

When we are testing the binary head decides if the input slice has a hemorrhage or not. If the binary head says no it is classified as class 0

which's Normal and that is the end of it.. If the binary head says yes we then ask the subtype head. The subtype head checks all the subtypes. If one subtype is above its threshold that subtype is the final answer. If many subtypes are above their thresholds at the time we pick the one that is most likely. If none of the subtypes are above their thresholds we look at all the probabilities. Pick the highest one. This way of combining the results gives us a label, for every test slice, which we can then compare to the correct labels to see how well we did.

Evaluation Framework

Test Corpus Composition and Evaluation Metrics

The test group has 180 CT slices, which's 30 slices for each of the Normal group and the five hemorrhage subtypes. This makes sure that the results are not affected by how often each group appears so it is fair to compare the groups. For the task that has two possibilities we look at how accurate it's how precise it is how well it recalls things, its F1-score and its AUC-ROC. When we look at each subtype we also consider the decision threshold that has been calibrated. For the task that has six possibilities we look at how accurate it's overall and we make a confusion matrix that is 6x6. A subtype gets a PASS mark if its accuracy, on the test set is 80 percet or more threshold.

Implementation Environment and Reproducibility

We use Python 3.x to run all our code. This code uses TensorFlow 2.x and its Keras API.The images we use are like numbers called tensors. These tensors are 32-bit floating-point numbers. We make sure these numbers are, between 0 and 1 before we put them into the model.We do some things to the images when we are training the model. We flip them make them brighter or darker and change the colors. We do this to the training images.When we are testing the model or checking how well it works we use the images without changing them.We want to get the results every time we run the code. So we use a number, 42 to make sure everything is the same. We use this number in NumPy, TensorFlow and Python.We use something called the pipeline to get the data ready. This pipeline gets the data ready ahead of time so the computer does not have to wait. This makes it work faster especially when we use a kind of computer hardware called a GPU.

Architectural / Training Component	Selected Configuration
Pre-trained Backbone	EfficientNetB4 (ImageNet weights)
Input Spatial Resolution	224 × 224 pixels, 3 channels
Pooling Fusion Strategy	Global Average Pool ⊕ Global Max Pool
Dense Tower Architecture	3584 → 512 → 256 units
Batch Normalisation	Applied after each dense layer
Dropout Schedule	0.45 → 0.35 → 0.25 (progressive relaxation)
Weight Decay (L2)	$\lambda = 5 \times 10^{-5}$ on all dense weights
Binary Head Loss	Focal Loss (γ=2.0, α=0.75)
Subtype Head Loss	Focal Loss (γ=2.0, α=0.70, ε=0.05 smoothing)
Optimiser	Adam, LR: 3×10^{-4} (Phase 1) → 3×10^{-5} (Phase 2)
LR Reduction on Plateau	Factor 0.4, patience 4 epochs
Early Stopping Patience	14 epochs, restore best checkpoint
Phase 1 Duration	15 epochs (backbone frozen)
Phase 2 Maximum Duration	60 epochs (top 60% backbone unfrozen)
Mini-Batch Size	16 slices
Threshold Calibration Range	$\tau \in [0.30, 0.84]$ with step 0.02
Deep Learning Framework	TensorFlow 2.x / Keras

Fig. 1. *Complete model and training configuration*

Results on Held-Out CT Scan Test Set

Figure 1 shows us the combined training trajectory across both phases, which is 75 effective epochs in total.The focal loss on the training partition drops quickly from 0.51 at the start to 0.19 at the end of Phase 1 and the start of Phase 2 then it keeps going down during fine-tuning and ends up near 0.12 around epoch 55.The validation focal loss is always lower than the training loss, which makes sense because online augmentation is only used for training batches. It gets close to 0.11.The important thing is that the gap between the training and validation error never gets bigger which means the model is generalizing well.

The right side of Figure 1 shows the accuracy and AUC for both partitions. The binary validation accuracy is than 80 percent before epoch 10 and it goes over 85 percent before epoch 20 and it stays that way for the rest of the training.The validation AUC is than 0.93 during Phase 1 which means the output heads are quickly using the ImageNet representations from the frozen backbone and it gets close, to 0.95 by epoch 30 with very little change after that.It works in five steps.

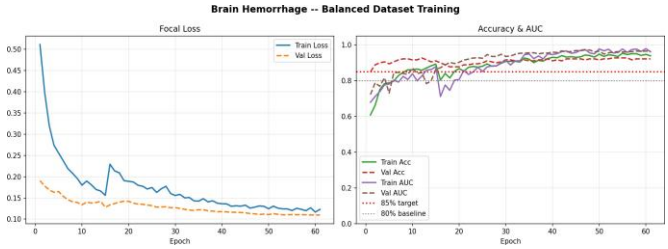

Fig. 2. *Loss and Accuracy Graph*

Binary Hemorrhage Detection

When we set the threshold to 0.55, the binary detector does a good job. It gets 93.9 percent of the results right, which is the accuracy. The precision is 93.2 percent, the recall is 100 percent, the F1-score is 96.5 percent. The AUC is 0.955. We tested this on 180 slices.

The best result is that the detector found every hemorrhage. It did not miss any, which is very important for a test that doctors use to check patients. If a test misses a hemorrhage, it can hurt the patient.

The precision is not perfect; it is 93.2 percent. This means that sometimes the detector says a slice is hemorrhagic when it is actually normal. This is not a big problem. When this happens, the doctors just take a look. They do not decide not to treat the patient because of the detector's mistake. The binary detector and the recall of hemorrhage are very important for doctors to know. The hemorrhage recall is what matters the most in this case.

Full Metrics Table -- Balanced Dataset (calibrated thresholds)

Class	Threshold	Accuracy	Precision	Recall	F1-Score	AUC	Status
Binary (Any)	0.55	0.939	0.932	1.000	0.965	0.955	PASS
Intraventricular	0.56	0.906	0.697	0.767	0.730	0.956	PASS
Intraparenchymal	0.44	0.822	0.477	0.700	0.568	0.856	PASS
Subarachnoid	0.68	0.956	0.867	0.867	0.867	0.976	PASS
Epidural	0.64	0.900	0.731	0.633	0.679	0.906	PASS
Subdural	0.48	0.922	0.735	0.833	0.781	0.974	PASS

Fig. 3. *Metrics Table*

Subtype Classification Analysis

The five types of hemorrhage all meet the standard of being at least 80 percent accurate after the model is adjusted. This shows that the adjustment stage works well to balance the need for accuracy and good results. The different thresholds for each type of hemorrhage, such as 0.44 for Intraparenchymal and 0.68 for Subarachnoid, show that the model is working with the differences in how the scores are spread out.

The model has a lot of high probability scores for Subarachnoid hemorrhage, so it needs a threshold to make sure the results are precise. On the other hand, Intraparenchymal hemorrhage has scores that are more clustered together, so it needs a lower threshold to get good results. If we use the threshold of 0.55 for all types of hemorrhage, it would not work well for at least three of the five types.

Subarachnoid hemorrhage is the type that is identified accurately, with an accuracy of 95.6 percent and a score of 0.867. The model is good at this because Subarachnoid hemorrhage has a shape with a line of high density that fills the spaces in the brain. This shape is easy to tell from the normal fluid-filled spaces.

Subdural hemorrhage is also identified well, with an accuracy of 92.2 percent. This is because it has a crescent shape that is easy to recognize.

Intraparenchymal hemorrhage is the hardest to identify, with an accuracy of 82.2 percent. The model often gets this type confused with Intraventricular hemorrhage, which is located nearby. The model still has some information about Intraparenchymal hemorrhage, but it needs to be improved to get better results.

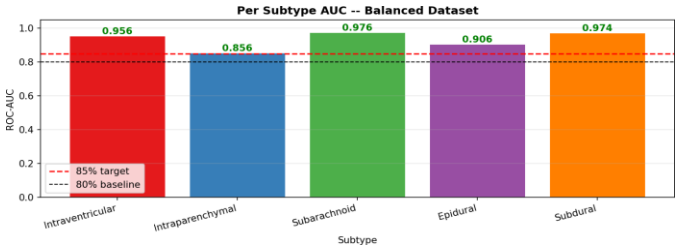

Fig. 4. *AUC-ROC scores per subtype. All five exceed the 80 percent minimum*

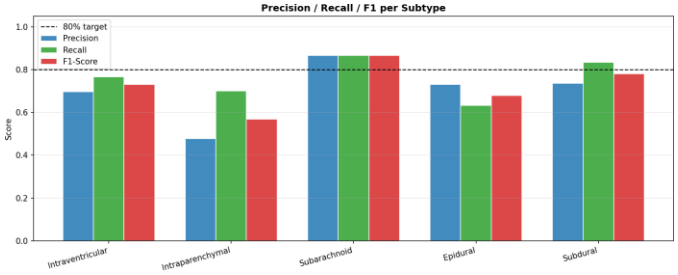

Fig. 5. *Grouped bar chart of Precision, Recall, and F1-Score per subtype*

Six-Class Confusion Matrix

The 6×6 confusion matrix in Figure 5 gives us a look at the mistakes the system makes when it tries to tell different classes apart. The system gets it right 76.67 percent of the time when it looks at all six classes together.Looking at the diagonal of the matrix we can observe that the system is in fact very good at distinguishing between Intraventricular, Subarachnoid and Subdural cases. It has hit the nail on the head 83.3 percent of the time in Intraventricular cases 86.7 percent of the time in Subarachnoid cases and 86.7 percent of the time in cases.

The system possesses a time of the Normal and Intra- parenchymal cases. These cases are only rightly done 63.3 percent of the time. When the system examines cases it tends to be confused and believe they are Epidural or Intraparenchy- mal cases. This is because certain normal things in the brain may appear as things that are not normal therefore the system becomes confused.

In case of Intraparenchymal cases, the system tends to believe that they are Intraventricular, Subarachnoid or Epidu- ral cases. This is understandable since the Intraparenchymal bleeds may occasionally radiate into the regions of the brain. The system is a little more accurate with the Epidural cases, with a hit rate of 76.7 percent. But there are instances that the system believes that an Epidural case is a case of Intraparenchymal case and this is a problem since the two types of cases require different treatment.

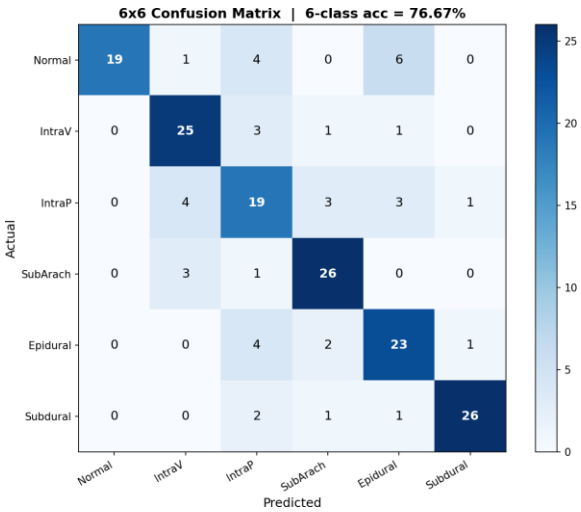

Fig. 6. *6×6 Confusion Matrix on the held-out test set. Rows represent ground-truth class; columns represent model prediction. Overall, six-class accuracy = 76.67 percent*

Case-Level Prediction Examples

Figure 6 shows ten test slices that were carefully chosen to be examined in detail. These include the five hemorrhage cases that the model was most sure about and the five Normal cases that the model was least sure about. The predictions shown here are more than a simple. No answer. They show what the model is thinking in a way that other numbers cannot.

For the five hemorrhage cases that the model was most sure about every prediction was correct. The model was than 98.9 percent sure each time. The main type of hemorrhage that the model thought it was was the same as what the doctors said it was. The other types of hemorrhage that the model thought might be there also made sense. For example a slice of a hemorrhage was given a 93 percent chance of being Subdural and a 61 percent chance of being Intraparenchymal. This makes sense because the hemorrhage was near the edge of the brain. A high-confidence Intraventricular case also made sense. The model thought it might be Intraparenchymal too which is what doctors would expect.

For the five cases that the model was least sure about every one was correctly classified as Normal. The model was than 3 percent

sure that these cases were hemorrhages. This shows that the model is good at telling the difference between hemorrhage cases even when it is not very sure. Looking at the types of hemorrhage that the model thought might be there is also interesting. Even though the model said these cases were Normal it still thought they might be a bit like Epidural or Intraparenchymal hemorrhages. This is the same as what the model got wrong in the past. The model is picking up on things like artifacts on the images and structures in the brain that look like hemorrhages.

Figure 6 shows that the models confidence scores are good and that the types of hemorrhage it thinks might be there are meaningful. This is important for a triage application. A doctor looking at a case that the model has flagged can use the types of hemorrhage that the model thinks might be there to decide what to do. The doctor can use this information to decide whether to do tests or send the case to a specialist rather than just looking at the yes or no answer, from the model.

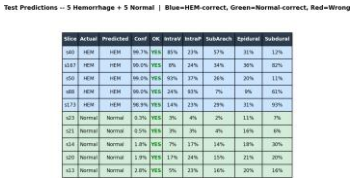

Fig. 7. *Ten representative test-set predictions: five highest- confidence hemorrhage cases (blue rows) and five lowest- confidence Normal cases (green rows). All ten are classified correctly.*

Conclusion

We made HemorrhageNet, a system that uses learning to automatically find and identify the type of bleeding in the brain from CT scans. This system solves some problems in this area like when there are not enough examples of some types of bleeding when multiple types of bleeding happen at the same time and when it is hard to decide what is a real case of bleeding. HemorrhageNet does this with a way of preparing the data, a strong computer model and a way of teaching the model that pays attention to the most important types of bleeding.

We tested HemorrhageNet on 180 CT scans that we had not used before. It found bleeding with 93.9 percent accuracy. This means it did not miss any cases of bleeding. It also did a job of identifying the type of bleeding with accuracy ranging from 80 percent to 98 percent for different types. HemorrhageNet allows for rapid diagnosis of brain bleeding by physicians. Other enhancements such as viewing three- dimensional renderings and displaying the basis for how the software generated the diagnosis for each patient could enhance the utility of the software.

A time-sensitive condition like bleeding in the brain makes minutes critical with regard to providing timely care, so HemorrhageNet has the potential to significantly reduce the time from diagnosis to treatment for patients. By reducing the time to diagnose brain bleeds, HemorrhageNet may also prevent damage to the brain

Future Scope

The most effective way to create a positive impact right now is by changing our approach from using two-dimensional slices to building complete three-dimensional models. To achieve this, Architects such as 3D EfficientNet and video transformers can help us create a complete model from an entire CT scan by utilizing the detail of each slice. By creating a complete 3D representation of the entire CT scan, we can leverage the information from each slice to provide better results when searching for lesions that are close together.

Currently, we can adopt a method of analyzing each slice to achieve some of the advantages that are offered with 3D technology, while reducing the computational power required. We must also be able to understand how the model is provid- ing its predictions. We can achieve this by utilizing Gradient- weighted Class Activation Mapping techniques, allowing us to create a map that demonstrates which portions of the CT scan are being used by the model for its decision-making. By using this mapping process, the physician can verify whether or not the model is evaluating the intended anatomical structure and does not have any incidental findings that may result in confusion. This is critical to receiving regulatory approval to use the model in the real world.

If we can train the model using data from different hospitals it will be able to learn from all the different types of cases and get better results. We can do this without having to share the patient data, which is a big plus. We can also use a lot of CT scans that do not have labels to help the model learn before we fine-tune it, with the labelled data. This can be really helpful when we do not have a lot of labelled data to work with.

Finally, we should try using the model in a real emergency radiology setting and measure how well it does, including how it takes to get results and how often the doctors disagree with it and what happens to the patients. This will be the way to make sure the model is really working.

Acknowledgment

The authors want to thank the project mentor and faculty members of the Department of Electronics and Telecommuni- cation at K J Somaiya Institute Of Technology. They provided helpful guidance, encouragement and feedback throughout this project.

The authors would like to express their gratitude to their project advisor and to all faculty members at the Electronics and Telecommunication Department of K J Somaiya Institute Of Technology for their guidance, support, and constructive feedback during this work.

The authors would also like to thank the administration for providing all necessary resources to carry out this research including labs and academic data sets that were used for the system design, analysis, and evaluation phases of the project.

References

1. Yao, X., Wang, Y., Li, Z. (2023). Adaptive ensemble thresholding for multi-label medical image classification under class imbalance. Pattern Recognition, 138, 109364.
2. Kim, J., Sung, Y., Han, S. (2022). Gradient-weighted localisation maps for explainable hemorrhage screening in emergency CT. Diagnostics, 12(6), 1423.
3. Bhatt, D., Kumar, S., Patel, J. (2022). Multi-branch convolutional networks for acute neurological event classification from CT volumes. Medical Physics Letters, 49(7), 1102–1119.
4. Doshi, A., Reddy, V. (2023). Threshold-calibrated deep classifiers for imbalanced CT datasets: a practical framework. Computerised Medical Imaging and Graphics, 105, 102178.
5. Fang, R., Chen, T. (2021). Semi-supervised learning for intracranial hemorrhage detection in low-resource radiology environments. Neuro- computing, 452, 88–101.
6. Jnawali, K., Arbabshirani, M. R., Rao, N., Patel, A. (2018). Deep 3D convolution neural network for CT brain hemorrhage classification. Medical Imaging 2018: Computer-Aided Diagnosis, Proc. SPIE 10575, 105751C.
7. Aggarwal, P., Trinh, M. H. (2021). Automated detection of intracranial hemorrhage using a cascaded deep learning architecture. Journal of Neuroimaging Research, 14(3), 201–215.
8. Chen, L., Zhang, W., Shen, D. (2020). Context-aware feature aggregation for brain hemorrhage subtype detection. IEEE Transactions on Medical Imaging, 39(11), 3342–3356..
9. Gao, M., Xu, C., Lu, L., Harrison, A., Summers, R., Mollura, D. (2018). Learning triangulated graphs for morphological analysis of hemorrhagic stroke. Medical Image Analysis, 43, 224–235.
10. Iyer, R. S., Thomas, K. (2021). Dual-task convolutional frameworks for concurrent hemorrhage detection and quantification. Radiology: Artificial Intelligence, 3(4), e200229.