# DEEP LEARNING FOR IMBALANCED IMAGE DATASET CLASSIFICATION

# DEEP CONVOLUTIONAL NEURAL NETWORKS FOR MULTICLASSIFICATION OF IMBALANCED LIVER MRI SEQUENCE DATASET

By ADITYA TRIVEDI, HBSc.

A Thesis Submitted to the School of Graduate Studies in Partial Fulfillment of the Requirements for the Degree Master of Science

Master of Science (2020) McMaster University
(eHealth) Hamilton, ON, Canada

TITLE: Deep Convolutional Neural Networks for Multiclassification of Imbalanced Liver MRI Sequence Dataset

AUTHOR: Aditya Trivedi
HBSc (Chemistry)
McMaster University, Hamilton, Canada

SUPERVISOR: Dr. Thomas Doyle

NUMBER OF PAGES: xi, 124

# Abstract

Application of deep learning in radiology has the potential to automate workflows, support radiologists with decision support, and provide patients a logic-based algorithmic assessment. Unfortunately, medical datasets are often not uniformly distributed due to a naturally occurring imbalance. For this research, a multi-classification of liver MRI sequences for imaging of hepatocellular carcinoma (HCC) was conducted on a highly imbalanced clinical dataset using deep convolutional neural network. We have compared four multiclassification classifiers which were Model A and Model B (both trained using imbalanced training data), Model C (trained using augmented training images) and Model D (trained using under sampled training images). Data augmentation such as 45-degree rotation, horizontal and vertical flip and random under sampling were performed to tackle class imbalance. HCC, the third most common cause of cancer-related mortality [1], can be diagnosed with high specificity using Magnetic Resonance Imaging (MRI) with the Liver Imaging Reporting and Data System (LI-RADS). Each individual MRI sequence reveals different characteristics that are useful to determine likelihood of HCC. We developed a deep convolutional neural network for the multi-classification of imbalanced MRI sequences that will aid when building a model to apply LI-RADS to diagnose HCC. Radiologists use these MRI sequences to help them identify specific LI-RADS features, it helps automate some of the LIRADS process, and further applications of machine learning to LI-RADS will likely depend on automatic sequence classification as a first step. Our study included an imbalanced dataset of 193,868 images containing 10 MRI sequences: in-phase (IP) chemical shift imaging, out-phase (OOP) chemical shift imaging, T1-weighted post contrast imaging (C+, C-, C-C+), fat suppressed T2 weighted imaging (T2FS), T2 weighted imaging, Diffusion Weighted Imaging (DWI), Apparent Diffusion Coefficient map (ADC) and In phase/Out of phase (IPOOP) imaging. Model performance for Models A, B, C and D provided a macro average F1 score of 0.97, 0.96, 0.95 and 0.93 respectively. Model A showed higher classification scores than models trained using data augmentation and under sampling.

# Acknowledgements

I would like to acknowledge my supervisor Dr. Thomas Doyle for his immense support, guidance and direction throughout the thesis.

I would like to thank Dr. Christian B van der Pol for his support, providing the image scans and suggestions in my research.

I would also like to thank Dr. Mohammadreza Heydarian for his recommendations and insight in the project.

Lastly, I would like to thank my family for their moral support throughout my MSc degree.

# Contents

# List of Figures

# List of Tables

# Chapter 1

# Introduction

## 1.1 Overview

Hepatocellular carcinoma (HCC) is the third most common cancer related cause of death. Hepatitis B and C are some of the common risk factors for developing into HCC and the risk of developing HCC is 2%-8% in cirrhotic patients[1]. Early diagnosis of HCC in patients is important for early treatment intervention. Annual new cases of HCC in North America are estimated to range from 500,000 to 1 million which leads to increased burden on healthcare (median cost of $176,456 per patient per year) [1][2].

Various imaging techniques have been used for diagnosis of HCC such as Ultrasound, Computed Tomography (CT) and Magnetic Resonance Imaging (MRI). Among these, MRI provides a more definitive screening tool (with no exposing radiation to the body) for HCC diagnosis [3]. Using multi-modal MRI (multiple T1 weighted, T2 weighted imaging), radiologists are able to apply the Liver Imaging Reporting and Data System (LI-RADS) to risk stratify a liver observation for HCC [4]. LI-RADS is a standardized algorithm endorsed by the American College of Radiology that aims to interpret and report liver examinations of patients at risk for HCC. It is used for population of adults older than 18 years of age [5]. A score category (Table 1) is assigned to a liver observation that ranges from benign to malignant observed in patients with chronic liver disease [5][6]. Gadolinium contrast agent is administered to capture imaging characteristics in various post contrast phases such as arterial phase, portal venous phase, hepatobiliary phase and transitional phase. These characteristic patterns are important for applying the LI-RADS score to a liver observation [6]. Thus, a radiologist provides a diagnosis from imaging characteristics relevant to HCC obtained from multi MRI sequence and phase information.

Table 1: LI-RADS Categories [4]

| **LI-RADS Category** | **Observation** |
|---|---|
| LR-NC (Not Categorizable) | Poor image quality |
| LR-1 | Definitely benign |
| LR-2 | Probably benign |
| LR-3 | Intermediate malignancy probability |
| LR-4 | Probably HCC |
| LR-5 | Definitely HCC |

Machine learning seeks to implement automation on tasks ranging from speech translation to image classification. Traditional machine learning algorithms require a feature extractor which extracts patterns from an image [7]. A sub class of machine learning represented as deep learning which is an improvement over machine learning methods do not require a hand-crafted feature extractor to learn patterns from a dataset. Over the past decade, an increase in interest towards deep learning has been fueled with optimization of computing resources such as GPU (Graphics Processing Unit) [8]. With the increase in availability of data the realm of deep learning has expanded to include novel techniques to outperform tasks performed by experts. Computer vision tasks are based on the idea of human visual recognition and have immense application in the field of radiology [9]. Advances in image recognition and classification have allowed deep learning techniques such as convolutional neural networks to not only extract features from an image but also determine category specific characteristics [10]. Convolutional neural networks (CNN) are a class of supervised learning architecture that have shown success in automating clinical decision making in medical imaging.

After going through different MRI image scans, radiologists look for imaging patterns that are characteristic for a particular disorder. This process can be physically and mentally fatiguing for the radiologist. However, these patterns can be captured by a deep learning model such as Convolutional Neural Network (CNN) given a labelled dataset of image scans to learn the patterns from. This learning can be compared to what is observed during training of a radiologist throughout their medical education with the difference being that deep learning models learn the boundary distinctions among different abnormality classes in short period of time [9][10]. To aid the radiologist in early diagnosis of HCC, deep learning models can be used to classify characteristic patterns observed from liver MRI scans.

Deep learning systems trained on large number of image dataset perform well with uniform distribution of samples among all classes [11]. Real world datasets, such as medical imaging, are most often imbalanced with the ratio of abnormal class being lower than that of normal class. This imbalance has shown to deteriorate classification performance of CNN models as the predictions are biased for the majority class [12]. Various methods of balancing the dataset have been explored in our work to design classifiers for sequence classification of liver MRI scans on an imbalanced dataset.

## 1.2 Motivation and Objective

The objective is to develop a deep CNN to classify liver MRI sequences from an imbalanced multiclass dataset. This deep learning model will classify the MRI sequence from images, with no sequence label, for further usage by the liver-cancer decision support. Data augmentation and random under sampling has been explored for the balancing of training data to reduce imbalance and bias towards majority class distribution of images. Sequence of a given MRI scan can provide vital information for assigning a LI-RADS score to a liver observation. Figure 1 provides a descriptive diagram for our contribution.

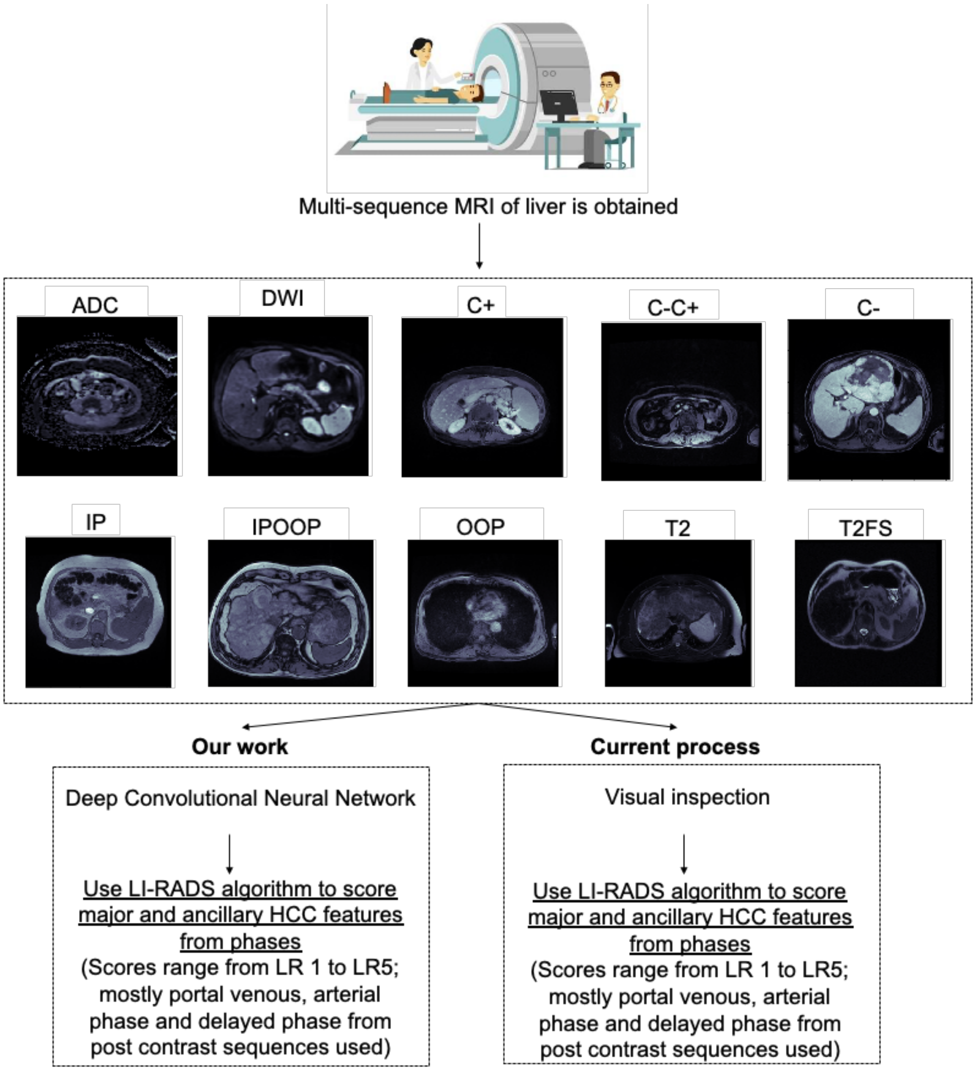

Multi-sequence MRI of liver is obtained
ADC
DWI
C+
C-C+
C-
IP
IPOOP
OOP
T2
T2FS
Our work
Current process
Deep Convolutional Neural Network
Use LI-RADS algorithm to score major and ancillary HCC features from phases
(Scores range from LR 1 to LR5; mostly portal venous, arterial phase and delayed phase from post contrast sequences used)
Visual inspection
Use LI-RADS algorithm to score major and ancillary HCC features from phases
(Scores range from LR 1 to LR5; mostly portal venous, arterial phase and delayed phase from post contrast sequences used)

Figure 1: Overview of our work contribution towards decision support of HCC using LI-RADS. The different MR image sequences captured for a patient are read by the radiologist to highlight characteristic liver observations for HCC.

After sequence information of a liver scan is taken into consideration by a radiologist, key imaging characteristics are combined to reach a LI-RADS score category. To automate this current workflow (Figure 1), deep CNN will classify a given image into one of the 10 sequence classes.

## 1.3 Thesis Organization

The flow of the thesis is as follows:

1. Background and Related Work

   This section is included to give the reader sufficient information on the background of the thesis and gaps in current work.

2. Methodology

   Methods and design of the experiments are included in this section.

3. Results

   Presentation of the results obtained based on the methodology undertaken.

4. Discussion

   Evaluation of the results are discussed in this section.

5. Conclusion and Future Work

   This section explores the future steps based on the lessons learned.

# Chapter 2

# Background and Related Work

This chapter provides an overview of relevant background and related work to support this thesis. The sections of this chapter begin with providing background on Hepatocellular carcinoma and the role of artificial intelligence in imaging. Furthermore, a comparison of machine learning frameworks and deep learning has been provided. In particular, deep learning framework such as convolutional neural networks have been explored. Lastly, different methods (algorithmic and data level) of classification of imbalanced datasets have been discussed.

## 2.1 Hepatocellular Carcinoma and Imaging

Hepatocellular carcinoma (HCC) is the third most common primary cancer of the liver and can be challenging to treat at a more advanced stage [13]. Diagnosis and treatment at an earlier stage is therefore critical to optimize health outcomes. Imaging exams such as ultrasonography (US), computed tomography (CT), and magnetic resonance imaging (MRI) are all frequently used to assess the liver in patients at risk of HCC [14][15]. HCC is the fifth common tumor type in Western societies, 80% of HCC cases develop in patients suffering from a cirrhotic liver [1][2]. Patients with chronic hepatic inflammation or non-alcoholic steatohepatitis are also at risk of developing HCC [16]. Family history of HCC, obesity and heavy alcohol consumption are other factors that increase chances of HCC in patients [17]. For primary surveillance, ultrasonography has been used though a more definitive imaging examination using multiphasic MRI and CT is then used to confirm the diagnosis [3].

Ultrasound is used for screening for HCC in high risk patients. Intravenously injected non-nephrotoxic microbubble based contrast agents serve as markers of blood in CEUS [18]. This technique shows higher sensitivity to detect arterial-phase hypervascularity and washout of HCC as compared to CT or MRI [19]. A study by Jang et al showed that the moderately differentiated HCC showed hypervascularity more often than poorly differentiated ones which hints that pathology of HCC is related to its enhancement patterns [20]. Deeply located (more than 12 cm) liver parenchyma lesions are difficult to diagnose using CEUS. Furthermore, detecting different lesions in same liver requires more than once administration of the contrast agent. Lastly, quality of CEUS scans are limited by bowel gas or large body habitus [21].

A multi-sequence MRI is also used to provide diagnosis of liver observations relating to HCC. In order to have fast acquisition of data, T1 weighted MR imaging is usually performed with requiring the patient to hold their breath while undergoing scanning. T2 weighted imaging can also be obtained using breath hold [22]. HCC confirmation using MRI is achieved using a multiphasic approach wherein diffusion weighted imaging (DWI) can provide increased detection rate [23]. There can be institutional variations with regards to the MRI sequences protocol though basic methodology is based on the Liver Imaging Reporting and Data System (LIRADS). T2 weighted imaging is less sensitive than DWI while non-enhanced T1 and T2 weighted are useful in focal liver disease characterization. DWI detects the Brownian motion of water molecules within the liver tissues and is similar in accuracy to contrast enhanced imaging [24]. Thus, DWI followed by contrast-enhanced imaging provides superior detection and characterization of liver observations [25]. Furthermore, DWI can be helpful in detecting small observations which otherwise are unnoticeable in contrast enhanced images [26]. Thus, a multi sequence approach provides for characteristic liver observations.

Gadolinium based chelating contrast agent is used to capture detailed lesion characterization in contrast enhanced MRI imaging. Arterial dominant, delayed, venous and interstitial phases of enhancement are usually captured. Then to obtain arterial nodule enhancement, unenhanced images are subtracted from contrast enhanced arterial phase. However, such method requires careful breath holds during patient scanning. It should also be noted that only 3-5% of administered gadolinium contrast agent is take up the liver cells thus most is excreted through the renal system [27].

By assessing imaging features of a liver observation on a multi-sequence MRI, radiologists can risk stratify the likelihood of HCC using LI-RADS. This requires the radiologist to assess multiple sequences to identify the presence or absence of major and ancillary features, which are used to assign a score between LI-RADS 1 (definitely benign) and LI-RADS 5 (definitely HCC) for each liver observation. LI-RADS has been endorsed by the American Association of Study of Liver Diseases (AASLD) and is becoming more commonly used in clinical practice around the world. LI-RADS helps radiologists to be consistent when applying a level of risk to a liver observation for HCC in high-risk patients [4] [28]. A LI-RADS 5 liver observation is highly specific for HCC and generally does not require tissue sampling prior to treatment.

Other imaging techniques have also been considered for diagnosis of HCC. Sequential imaging wherein inconclusive MRI findings followed by Computed Tomography (CT) scan for HCC were found to be costly and left about 20% of cases undiagnosed [29]. Ultrasound guided biopsies of the liver are an invasive technique that use normal greyscale US to increase target tissue selection. Since only 33% of HCC nodule characterisation meet the imaging diagnostic criteria laid out by American Association for the Study of Liver Diseases [30], 50-70% of cases require a liver biopsy to confirm the HCC diagnosis. Thus, liver biopsies are useful when there are conflicting observations from imaging results for small focal lesions [31]. However, liver biopsy can cause pain and discomfort and could possibly show a seeding risk [32].

Hence, as compared to ultrasound guided biopsy, MRI stands out as non-invasive tool. Contrast administered post contrast sequences such as C+, C-C+, C- provide further information for the HCC enhancement patterns. Arterial phase (AP) liver MRI capture is crucial in applying LI-RADS algorithm which helps to identify major observations related to HCC diagnosis; it is characterized by complete enhancement of hepatic artery [5]. It is further divided into early (EAP) and late arterial phase (LAP). In the early Arterial phase, no enhancement of portal vein is observed while the late Arterial shows enhancement of the hepatic veins. HCC enhancement is usually maximum in the late Arterial phase [6]. Thus, Arterial phase shows post contrast injection observations relating to HCC. Delayed phase (DP) also represents observation from post contrast imaging and is obtained after Portal Venous phase. Portal Venous phase (PVP) provides full enhancement of portal veins and the liver parenchyma [33]. Transitional phase (TP) is acquired before Hepatobiliary phase (HBP) and characterized by high liver parenchyma enhancement or signal intensity. The HBP is obtained last (20 minutes after contrast injection) as the contrast agent is getting excreted through the biliary system [6]. In HBP, the liver blood vessels appear hypo intense than the liver parenchyma. Vernuccio et al [34] have evaluated diagnostic performance of HBP hypo intensity for HCC diagnosis. Through a multi institution study they report 80% specificity for HCC when HBP was used to stratify the liver observation to a LI-RADS score. Thus, along with MRI sequence information, phase of enhancement in post contrast sequences is also useful in application of LI-RADS.

## 2.2 Artificial Intelligence in Medicine

On an average, radiologists have to report one image in less than five seconds and with the increase in workload there is a possibility for errors in diagnosis [9]. It is estimated that implementing clinical decision support tools to supplement radiologist's workflow can not only improve efficiency but also reduce the chance for possible misdiagnosis [35].

Artificial Intelligence is a broad term that encompasses machine learning and deep learning methods which aims to learn patterns or boundary distinctions among categories from a given data distribution. Feature extraction methods used in machine learning require manual preparation for the images to learn the targeted features of interest. Various approaches have been implemented for feature learning however, deep learning has emerged as an approach that reduces reliance on manual selection of feature extraction methods [9].

The following sub sections will explore different methods in machine learning and deep learning that have been applied in different imaging datasets.

### 2.2.1 Deep Learning

Various machine learning methods for feature extraction and classification along with deep learning methods have been explored and summarized in Table 2.

Table 2: Summary of machine learning and deep learning methods

| Study Reference | Objective | Method | Findings |
|---|---|---|---|
| Meng et al [7] | Two-class tumor classification on dataset of 100 MRI images | SVM (Support Vector Machine) | Average accuracy of 83% was reported |
| Levman et al [38] | Breast lesion classification from 94 MRI scans | SVM classifier for breast cancer using DCE-MRI | Area under ROC curve (classification measure) was 0.74 |
| Faria et al [39] | Feature extraction from brain scans | PCA (Principal Component Analysis) | Accuracy of 88% in distinguishing Primary Progressive Aphasia from normal scans |
| Evangelia et al [40] | Brain tumor detection in multi-sequence MRI dataset | SVM | High computational cost for multiclassification |
| Siddiqui et al [41] | Feature extraction for brain MRI diseases | Discrete Wavelet Transform | Overfitting on a small dataset and manual tuning of wavelet coefficient was required |
| Qureshi et al [43] | Feature extraction for ADHD classification | Recursive Feature Elimination (RFE) SVM | Based on ranking list and shows good results for metabolomics |
| Zhang et al [45] | Feature extraction and tumor classification | PCA was used for feature extraction and SVM for tumor classification | Requires careful tuning and selection of the region of interest that needs to be extracted using PCA |

| | | | |
|---|---|---|---|
| | | from three MRI sequences | |
| Sun et al [47] | Image classification (Liver CT images) | Comparison of SVM to LDA (Linear Discriminant Analysis) and PCA for classification tasks | SVM showed higher performance (72% accuracy for binary classification of liver CT images) |
| Li et al [48] | Brain glioma feature classification | Linear SVM | 88 % accuracy reported though expert intervention was required to select features from image scans |
| Yasaka et al [51] | Classification of five types of liver masses from CT image scans | Deep Convolutional Neural Networks (CNN) | The classification accuracy reported was 95% without any manual feature selection |
| Nawaz et al [54] | Classification of breast cancer into benign and malignant | DenseNet CNN | An accuracy of 95% was reported using the transfer learning approach |
| Lakhani [55] | Tuberculosis identification from chest radiographs | Deep CNN | The classifier achieved an accuracy of more than 90% |
| Mohsen [56] | Brain tumor classification from T2 weighted MRI scans | Deep Neural Network with seven hidden layers | DNN achieved an accuracy of 91% over machine learning techniques such as K-Nearest Neighbors and SVM |
| Noughci et al [57] | Multi-classification of brain MR image sequence | GoogleNet and AlexNet DCNNs | Both architectures achieved an accuracy of at least 90% |

Computer-aided diagnostic classification has been used to expedite early detection of liver cancers using US [36][37]. Machine learning techniques which learn the association between the input and output from a labelled dataset are called supervised learning algorithms. Supervised machine learning techniques such as support vector machines (SVM) have been used to classify cancers using MRI and CT scans. A two-class tumor classification with Radial Basis function (RBF) using only T2-weighted MR images achieved an accuracy of 86% [7]. Studies have also explored using SVM techniques for brain and breast cancer classification. Dynamic contrast enhanced magnetic resonance imaging (DCE-MRI) were used to classify breast lesions as either cancer and benign wherein cancer was further separated into two classes (non-invasive and invasive) leading to a three-class classification achieved using SVM [38]. Using T1- weighted images, Faria *et* al showed that principal component analysis (PCA) can be used as a feature extractor for brain MRI scans [39]. With a multisequence MR image dataset of brain tumors in adult population, SVM with automated feature extraction was utilized for both binary and multiclass classification which gave an accuracy of 85%. Feature ranking and feature subset selection method used in this study cannot be applied to high dimensional classification tasks due to higher computational cost [40]. Discrete wavelet transform (DWT) used to extract features for multiclass brain MRI disease classification with SVM classifier showed more than 85% accuracy [41]. However, it should be noted that the dataset used in the DWT type method was small (less than 500 images) which could have let to overfitting and that DWT requires careful analysis of wavelet coefficient [42]. Extracting features from brain MRI scans for Attention Deficit Hyperactivity Disorder

(ADHD) types classification using Recursive Feature Elimination (RFE) SVM method provided 60% multiclass accuracy while showed higher accuracy for binary classification [43]. The RFE-SVM works by using a ranking list and works good for metabolomics since there is a need for noise reduction to accurately describe a signal [44]. A multi-kernel SVM using T2 weighted, Proton Density (PD) and Flair (Fluid Attenuated Inversion Recovery) MRI also employed PCA as a feature extractor to identify tumor classes from the three MRI sequences downstream [45]. Other methods such as Linear discriminant analysis (LDA) have also been used to classify breast MR images [36][46]. Compared to PCA and LDA, SVM has been shown to provide higher accuracy for image classification tasks [47].

Conventional machine learning techniques, such as SVM, require careful design of feature extractor that transforms the image scans into feature vectors which are fed to the classifier [10]. Training linear SVM on descriptive features as well as clinical grade based on the brain glioma dataset required expertise of neuroradiologist to define the features to be used and achieved an accuracy of 88% [48]. This has been the primary limitation of using SVM classifiers for big image dataset classification tasks.

Artificial neural networks (ANN) mimic the biological phenomenon through which humans learn and coordinate themselves. They have been used in medical image classification and provide better performance than conventional machine learning approaches [49]. ANN consist of interconnected neurons that communicate with each other to learn associations over the input features to determine respective class outputs. Deep neural networks are an extension of ANN with more layers of neurons [49][50].

Alternatively, deep learning techniques, such as convolutional neural networks (CNN), have the ability to self-learn features over large data [50]. Image classification tasks require robustness to variations in images such as position, orientation of the object, brightness or contrast of the image. Thus, the input-output function needs to be insensitive to changes in input image. Using a SVM would hence require micro-tuning of the feature extractor for image preprocessing tasks [51]. Compared with feedforward networks, CNNs require fewer parameters and connections making them easier to train. However, to train on large image sets with complex features, it requires GPU power to achieve high performance [52][8][53]. Yasaka et al used Deep CNN (DCNN) to classify five types of liver masses from dynamic contrast enhanced CT images and achieved an accuracy of 84% with minimal preprocessing. DenseNet CNN architecture developed initially for the ImageNet classification challenge, led to a 95% accuracy rate for classifying breast cancers into benign and malignant (sub-classes) using transfer learning [54]. Deep learning has also achieved high classification accuracy on chest radiography. DCNNs trained using a CUDA - enabled Nvidia Titan X 12 GB GPU to identify tuberculosis from chest radiographs achieved higher than 90% accuracy [55]. Classification performance of brain tumors (glioblastoma, sarcoma, metastatic bronchogenic carcinoma) from MRI (T2-weighted) images using DNN (Deep Neural Network) with seven hidden layers was compared with K-nearest Neighbors (KNN), SMO-SVM and LDA. It was found that DNN achieved higher accuracy as compared to other methods with less tuning for image preprocessing [56]. Nouguchi *et al* achieved an accuracy of more than 90% to classify six different MR image sequences (T2WI, T2*WI, FLAIR, DWI, ADC, TOF-MRA) using the GoogLeNet and AlexNet architecture (DCNNs) with a 10-fold cross-validation [57].

## 2.2.2 Convolutional Neural Network (CNN)

Images seen by a human eye are easily interpreted and classified into a categorical object. In a field of random objects, the human eye is able to distinguish the object of interest from the rest very easily. This approach is explained by the Spotlight model of attention [58]. Computer vision tasks are based on a similar approach which aim to mimic the innate human tendency to recognize and infer objects around them. Thus, computer vision tasks immensely depend on the features that it extracts from given pixels of an image which are later used for inference.

Convolutional neural networks (CNN) are an excellent class of feature extractors that have shown translational invariance [57]. CNNs learn spatial features from an image ranging from both high- and low-level patterns. The CNN method was implemented by Lecun et al. using two convolutional operations for document and digit recognition [59]. For CNNs to work with raw images a preprocessing is usually undertaken. This Preprocessing may include normalization, zero centered or same pixel sizes. From the input image pixels, the linear convolutional operation extracts the features which are then subsampled (Average

Pooling or Max Pooling). This is then fed to the next convolutional layer which further extracts key features making the hierarchy complex. A feature can be defined as the characteristic component of the object such as the wings of an airplane or eyes in a human face. Thus, on an input image local receptive field features combined with consequent layers identify the high level features of the image [59].

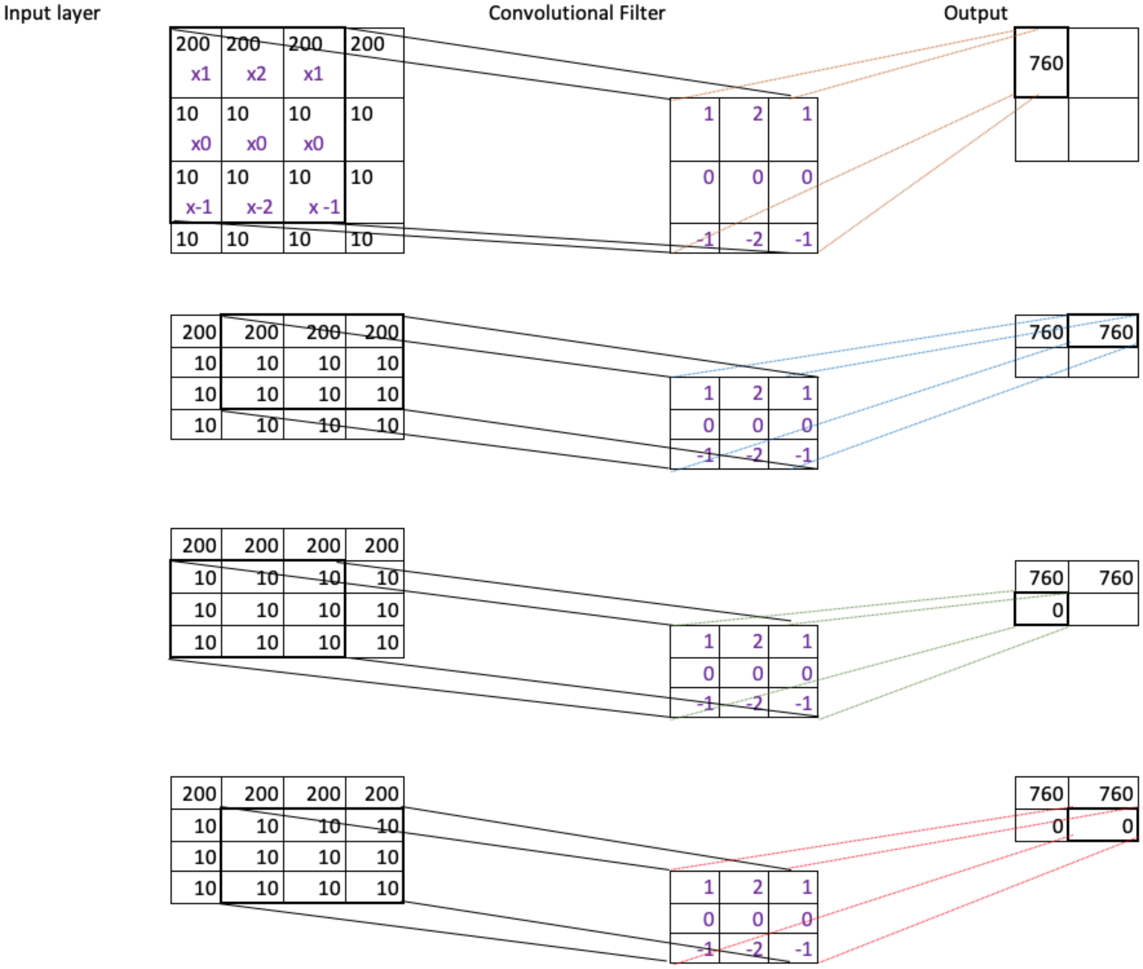


Figure 2: Convolution of a (3x3) filter on a sample input image (4x4x1) with a stride length of 1. Image pixel intensities assumed to be in range 0-255 where we see only intensities of 10 and 200 for this simple example.

From Figure 2, it is observed that given an input dimension of an image, a convolutional kernel will search for features on the input and will provide an output layer (matrix). This output layer then acts as the input for the second convolutional operation. Figure 2 shows the working of Sobel horizontal edge-detection filter on an input image. Similarly, different types of filter can be used depending on the classification task.

Figure 2 also shows that the operation of convolutional layer on the input image is not regular matrix multiplication of the two. The output is obtained by taking the sum of the point-wise multiplication of the filter elements to the underlying elements of the image. Below is a description of the formula used to calculate the shape of the output layer given an input dimension and convolutional kernel size [59].

Let the height of the image dimension be '$h_1$', width be '$w_1$'. Then the output dimension ($h_2$, $w_2$) is calculated by:

$$h_2 = \left(\frac{h_1 - C_h + 2P_h}{Stride\ length}\right) + 1$$

$$w_2 = \left(\frac{w_1 - C_w + 2P_w}{Stride\ length}\right) + 1$$

where ($C_h$, $C_w$) represents the convolutional kernel size and ($P_h$, $P_w$) represents padding.

For the image input in Figure 2, the dimensions are 4x4x1 for the grayscale image. The height is 4 and width is 4 as well. The CNN filter is 3x3 and the stride length is 1 with no padding. Hence using equation 1 and 2 we get:

$$h_2 = \frac{4-3}{1} + 1 = 2$$

$$w_2 = \frac{4-3}{1} + 1 = 2$$

Thus, the output dimension obtained will be of the shape 2x2 which is observed in Figure 2. This will be the input layer for the next convolutional operation in the network.

## 2.3 Imbalanced Dataset Classification

Clinical diagnostic datasets, similar to fraud detection datasets, inherently possess class imbalance. The imbalance refers to the underrepresentation of the minority class(es) as compared to the majority class. This occurs in both binary and multiclass classification problems [37][60]. Skewed class distribution in a dataset is inherently observed in image recognition, cancer classification, medical disease diagnosis and computer security tasks [60][61]. Medical data contains intrinsic imbalance while extrinsic imbalance is dependent on the data collection or storage methods [11]. Imbalanced class distribution causes over estimation of the classifier's accuracy and there is an increased probability of predicting the majority class [62]. According to Japkowicz, linearly separable non-complex classifications are unaffected by class rarity, though as the problem becomes complex (e.g., CIFAR-10 vs MNIST) sensitivity is reduced [12]. High accuracy and low error rates are misleading while evaluating model performance on imbalanced data since it is dominated by the majority class. Thus, other metrics, such as sensitivity and specificity, obtained from a confusion matrix are required to accurately evaluate classification performance. Receiver Operator Curve (ROC) alone also does not provide accurate measure of imbalanced classification performance [63].

Class imbalance and bias towards majority class can be reduced by three different approaches:

1. The first is at the dataset level, which involves either random under-sampling the majority class or random over-sampling the minority class. Under-sampling reduces the over representation of majority class and reduces bias while over-sampling involves increasing the number of minority class samples either artificially or by acquiring more data [64]. Synthetic Minority Oversampling Technique (SMOTE) has shown to be effective as compared to under-sampling though prone to overfitting, and also leads to an increased training time for larger image datasets [65].
2. Cost-sensitive learning is another method at the algorithm level that improves overall classification accuracy. This reduces false negative rate with goal of reducing the cost function value to a minimum [66].
3. Class weighting is another technique which is used to assign class weights to the different classes with the minority class receiving increased weighting. This allows the model to focus on the minority class without altering the original dataset. Since it requires careful examination and expertise in correctly assigning the class weights it is difficult to optimize over large datasets [67].

The next subsection will provide more examples on approaching imbalanced classification for both imaging and non-imaging data. Based on the above points dataset modification and algorithmic approach have been discussed. Sections 2.3.1 and 2.4.1 explain augmentation and sampling methods specific to this thesis.

## 2.3.1 Dataset Modification

### One-Sided Selection

One-sided selection used by Kubat and Matwin to balance two class dataset required removal of redundant, borderline and noisy samples from majority class that potentially do not affect the classification scores [68]. A limitation of this approach is that it requires careful selection of samples that have to be left out and not readily scalable to large datasets.

### Neighborhood Cleaning Rule

Neighborhood cleaning rule (NCR) utilizes selective under-sampling method to balance the dataset. This often results in improved sensitivity at the cost of reduced specificity [69]. NCR is an extension of one-sided selection that includes more selective removal of samples from majority class. In this technique, data noise in the neighborhood of the minority classes are removed. Emphasis on data cleaning methods prior to data reduction was more relied upon in NCR [70]. Heterogenous value difference metric was used by [54][55] over the Euclidean distance metric used in the one-sided selection method. With Edited Nearest Neighbor rule approach, NCR provided better accuracy as compared to the one-sided selection method for multi-class datasets while rates of true positives were insignificant

[70]. A limitation of this method points to the inefficiency of applying to datasets that have large number of samples as with increased number of samples, data preprocessing and reduction become very sensitive and requires expert intervention.

**Data Level Approach**

The CIFAR-10 dataset contains 6000 images in each class and hence is a balanced distribution among the classes. Evaluating the performance of CNN in classifying 10 classes in the CIFAR-10 dataset, [71] introduced artificial imbalance in the dataset and compared the classification performance of their classifier. It was observed that the distributions with most imbalance among classes (14% of total images were represented by airplane class) provided worst accuracy (10%) while the original balanced dataset provided 73% accuracy. Random oversampling (duplication of minority class images) performed on the underrepresented classes in the imbalanced distribution yielded increased performance of the classifier (73% for both) similar to the balanced distribution.

Sampling methods such as under sampling of majority class and over sampling of minority class improves the class distribution from the original imbalanced dataset. Random under sampling and over sampling approach allows to reduce the degree of imbalance within the modified training set [11]. Depending on the dataset, optimal level of under sampling or over sampling could be performed to reduce the imbalance. Compared to uniform balance it has been observed that performing under sampling on majority class only or minority class oversampling to reduce class imbalance could provide similar results to that of balanced distribution [64] [72] [73].

Using MNIST, CIFAR-10 and ImageNet datasets Buda et al [72] used CNN to classify images. Varying ratios of class imbalance was introduced in the training sets of the three datasets. Model parameters and architecture was kept constant throughout all three datasets. It was observed that the method of thresholding (adjusts decision threshold of output probabilities and utilized during test phase of a classifier) on the CNN outputs did not perform superior to random oversampling method of the minority classes for MNIST and CIFAR-10 datasets. Additionally, they report that oversampling did not cause overfitting for their predictions. Furthermore, Buda et al suggest that for imbalanced datasets using CNN oversampling should always be performed so as to reduce imbalance within the distribution prior to training.

## 2.3.2 Algorithmic Approach

**Learning Classifiers**

Learning classifier systems represent the use of rule-based type of learning which can be used in imbalanced datasets classification [73]. Investigation of performance of training data resampling on various imbalanced datasets with learning classifier system based on rule-based learning (reinforcement learning and supervised learning) provided an average accuracy of 65.91% and 68.23% for reinforcement learning and supervised learning systems respectively for small datasets. Using this rule based approach with under sampling and oversampling of imbalanced datasets it was observed that compared to over-sampling under-sampling led to poor classification performance [74].

### Transfer Learning

Marine plankton dataset (WHOI-Plankton) [75] consists of imbalanced distribution having 100 classes with only 5 classes representing more than 80% of the dataset. CNN classifier's performance (with and without transfer learning) trained using this dataset was compared with data sampling techniques such as random reduction of overrepresented data to balance the dataset [76]. Transfer learning with CIFAR10 CNN model was implemented after reduction of plankton images from majority class and provided an F1 score of 0.308 while a score of 0.177 was observed for the original imbalanced image distribution. The data reduction technique increased the classification accuracy of the minority classes while also increasing the overall accuracy of the pre-trained CNN classifier. Furthermore, they also experimented data augmentation techniques such as rotation, scaling and translation which improved the classification score (F1 score of 0.312).

Pouyanfar et al, have explored the use of real-time data augmentation i.e. transform image batches for each training step in multi-class dataset [77]. Transfer learning was achieved using the InceptionV3 model using the Keras ImageDataGenerator for data augmentation (horizontal flip, rotation, shear). It was observed that the model with augmentation performed better (F1 score of 0.553) than the imbalanced distribution (F1 score of 0.432).

Comparing the results from [71], [76], [77] it was inferred that random over-sampling and data augmentation can improve classification performance. Under-sampling requires removal of majority class representation and can reduce training time but reduction in majority data might require careful intervention so as not to remove representative data for that class distribution.

**Modifying Model Loss Function**

Impact of modifying loss functions during the training phase has also been explored to overcome class imbalance. Mean Square Error (MSE) is the most common loss function used in deep learning tasks, Wang et al [78] experimented with different loss functions (Mean False Error and Mean Squared False Error) to evaluate model performance. MFE computes error on the samples by taking average sum of the error from two classes. MSFE improves upon MFE since MFE loss is insensitive to positive class error rate and reduces classification accuracy for the majority class while focusing on the minority class [78].

Using the standard CIFAR-100 dataset, the super classes were further combined into three different datasets to artificially introduce imbalance (20%,10% and 5%). Wang et al observed that the classification performance was severely affected when the imbalance between the two classes was higher (5% of minority samples as compared to the majority ones). As compared to the MSE, the classification performance using MSFE and MFE reported minor improvement from baseline (had similar F1 scores). Additionally, Focal loss used by Lin et al [79] was a similar technique to reduce weighting on majority representation of samples. Their approach builds on the limitation of minority class gradient of the work in [80]. Object detection using R-CNN utilizing a two stage approach often encounter class imbalance [81]. Within the two stage object detector approach, the first stage identifies the desired objects of interest and then the second stage can use hard example mining to perform sampling heuristics [82]. Since the first stage has to look for object of interest, there was an imbalance ratio of background samples to the object to be detected. The focal loss method developed by [79] aimed to improve the efficiency of the one-stage detector by easily integrating into the existing model architectures. Results from using this focal loss method in image classification task of imbalanced architecture dataset by [83] for the decomposed classes (from six to three) revealed higher accuracy (83%) when the down weighting hyperparameter (increases focus on negative samples) of the loss function was set to zero. This value was similar to that obtained by the standard Cross Entropy loss which is a standard loss function for multi-class classification problems [84].

In binary classification of imbalanced datasets, [80] used modified backpropagation (descent vector) to reduce the error rate of minority class and improve convergence. As they report, this method was not extendible to multi-class problems due to very low error convergence rate for underrepresented classes in such classifications.

Based on the evidence from above works, it was observed that class imbalance can be improved prior to training and techniques such as under sampling prove to be less effective as compared to oversampling the minority class which led to less overfitting. Furthermore, it was also found that accuracy cannot provide a robust measure of the classifier's performance in a multi-class dataset. Metrics such as F1 score along with precision and recall scores should be used to assess model performance.

The next section will explore the use of data augmentation methods to balance imbalanced distribution of data.

## 2.4 Data Augmentation Methods

Rotation, translation, zooming or shearing are the most common form of data augmentation methods that have been applied to image classification with imbalanced datasets. These methods introduce new images in the dataset which are used during training of CNN models [85]. Since CNNs are not viewpoint equivariant, geometric transformations are useful for CNNs as they make it invariant to image positioning and orientation.

Transformations such as changing the color, contrast and brightness of the images is another technique to increase the number of samples in the dataset and prevent overfitting [86]. Comparing color transformations of images to that of rotating, flipping and cropping [86] report that CNN model accuracy increased when geometric transformations were used. Simple geometric transformations are not computationally intensive and are easy to implement on the training set.

### 2.4.1 Simple Image Augmentation Methods

Using the Caltech 101 (9146 images with 101 categories) and Pascal VOC (10,000 images with 20 categories) datasets, [87] evaluated image augmentations such as flipping and cropping on CNN model performance. Mean accuracy reported by [87] shows that model performance increased when cropping is combined with flipping. Primary limitation of their study was that only mean accuracy was used and hence other metric such as F1-score for a multi-class problem is required to validate their results.

Using the Galaxy Zoo dataset consisting of 61,578 images in the training set and 79,975 images in the test set, [88] explored the use of rotation and cropping to augment the training images in a small dataset and to increase robustness of the CNN model. Since rotating the images does not affect the classification based on morphology, random rotation of images in the range of 0° to 360° was performed. Other geometric transformations such as flipping images horizontally was also used. Furthermore, regions of interest were also cropped followed by random rotation to provide different viewpoints of each image. As reported by [88] their approach provided a score with an RMSE (Root Mean Square Error) value of 0.075 along with high recall (0.7) and precision (0.8) scores without averaging predictions from different models trained on the same dataset. Lastly, [88] report that geometric transformations provided better results as compared to photometric transformation such as increasing or decreasing brightness of images.

In a multi-class (18 categories) balanced fruit dataset, [89] have compared CNN classification performance with (3600 images with 200 images per class) and without augmented training set. The small fruit dataset consists of only 1800 images in the training set which were augmented by rotation (range of -15° to 15°), gamma correction, scaling and introducing noise in images to yield a balanced set of 64,800 images in total. Test set results show that CNN accuracy, sensitivity and precision scores improved after using the augmented images. Accuracy increased by almost 4% for the data augmentation approach while non-augmented approach had 86% accuracy.

Using CNN as a feature extractor for Synthetic Aperture Radar (SAR) images, [90] have explored data augmentations such as translation, speckle noise and changing pose (rotation) of images. Translation of images was performed to make the CNN robust to alignment of target features in the image. A set window (30 x 30 pixels) was defined within which defined the maximum shift along x and y axis. Speckle noise is inherently present in SAR images and reduces the ability of human observer to distinguish the target feature from background [91]. Ding et al [90] found that there was no significant difference in test accuracy when CNN was trained separately on different augmentations or combined augmentation. As reported by [92], when noisy images were used in the training set model accuracy was lower than that of image rotation and translation. Mean accuracy of 50% was observed for the speckled noisy image set while more than 85% for the translated set [91].

Hussain et al [92] compared different augmentation approaches to increase validation accuracy on mammography dataset. Augmentations such as horizontal and vertical flip, shear, rotation, Gaussian noise injection, scaling and blurring images were evaluated. The dataset consisted of balanced distribution between normal (1650 images) and non-normal (1651) mass samples. Images were cropped to reduce its size before and after augmentation to reduce number of parameters to train. VGG-16 CNN was then used on the augmented 15,673 images. Hussain et al report that validation accuracy suffers when noisy images are added to the training set. Simple geometric transformation such as flipping images gave more than 85% accuracy as compared to 78% accuracy prior to augmentation.

### 2.4.2 Algorithm Based Image Augmentation

Wang and Perez, explore standard geometric augmentation techniques (horizontal flip, cropping and rotating images by 45°) and augmentation using CycleGAN [93]. Generative adversarial networks (GAN) use a discriminator and generator network to artificially increase the training images. The generator network generates images based on the sample images while discriminator compares it to the original image [94]. Using the Dogs vs Cats and Dogs vs Goldfish dataset, [93] evaluated the performance of models trained on non-augmented and augmented training sets. It was observed that traditional augmentation (computationally less intensive) provided similar accuracy to that of network trained using images generated by GANs. Eighty nine percent validation accuracy was observed for the Dogs vs Goldfish data while 77% for Dogs vs Cats data when traditional augmentation was used. They also evaluated the use of neural augmentation for the MNIST dataset. Neural augmentation or learning augmentation utilizes two images from each class to generate a new image and then the original image and new image are fed to the next layer increasing the number of training samples available. However, [93] report that using neural augmentation and GAN do not produce significantly different classification accuracy and are more computationally expensive than the geometric transformations.

Cubuk et al [95] devised AutoAugment that learns augmentation strategies from the given dataset. Using various imaging datasets (CIFAR-10, CIFAR-100 and ImageNet) they have compared classification performance of the AutoAugment method. Their approach uses Reinforcement Learning to determine possible augmentation approaches (both geometric and photometric transformations) that give higher validation accuracy. A recurrent neural network is used to determine possible augmentation strategies (translation, color variations, sharpness, rotation and inversion) which are reinforced based on the validation accuracy received from the set model architecture. Similar approach was used by [96] but using GANs.

Cubuk et al also experimented using AutoAugment transfer approach which involves using the learned strategies from dataset such as ImageNet or CIFAR-100 and apply them to new datasets such as Caltech-101. They report that their transfer approach led to 2% reduction in error rate from baseline models though it should be noted that the size of training images had to be reduced. AutoAugment cannot perform on the whole dataset and requires a core sub sample for the training set on which the search algorithm works best. From the original training set (~550,000 images) of SVHN (Street View House Numbers)  dataset 73, 257 images [97] were chosen for the core training set on which AutoAugment approach provided geometric transformation as the most common augmentation strategy over photometric transformation. An average reduction in error rate of 0.4% (before augmentation test error was 1.5%) was observed for the SVHN dataset after training on augmented set which is not significant given the increased computational time (5000 GPU hours on CIFAR-10) [98] spent on augmentation search.

To overcome the challenge of class rarity in brain MRI datasets, [99] have used GAN to synthetically augment abnormal brain MR images. Using pix2pix GAN, they were able to generate T1 weighted, T2 weighted and FLAIR images and also control the location of tumor on the brain scan. The ADNI (Alzheimer's Disease Neuroimaging dataset) is a small dataset with 3000 images and thus due to its small size an approach involving GAN was used [100]. However, it should be noted that images had to be cropped and resized to a smaller size for the ADNI, focusing on the region of interest so as to reduce the time taken to generate the synthetic image.

Overall, it has been observed that simple geometric data augmentation methods such as flipping, rotating or translating the images can help to artificially increase the class balance as compared to data augmentation using GANs or AutoAugment that require increased computational time and resources.

## 2.5 Model Architecture Design and Evaluation

This section focuses on previous work on deep CNN model designs along with hyper parameter tuning which improves classification performance. Multiple architectures have been explored for imaging datasets.

### 2.5.1 Convolutional Neural Network Architectures

Deep learning models can converge and scale well when training samples are increased. Very deep learning models as used by [101] show that model error rate converges faster when more convolutional layers are used. More than 10 convolutional layers were used with 3x3 as the kernel size for all layers. The dataset was imbalanced facial recognition images and the model performance was evaluated on the EmotioNet challenge. They have compared classification error rates for deep (18 layers) and shallow (6 layers) network architectures. Max pooling was used after the first convolutional layer with a stride of 2 and size of 2x2. They report faster error convergence rate when deeper network was used. Furthermore, their 10 layered CNN network provided higher F1 score (0.641) for the majority class as there was no attempt to balance the dataset prior to training.

Several CNN architectures have been designed with optimizations to parameter, variation of regularization techniques and stacked layering [102]. LeCun et al developed the first CNN architecture for digit classification [103]. It included five convolutional followed by pooling layers. AlexNet was later developed overcoming the shortcomings from LeNet to be applicable to image classification tasks [104].

### Hyperparameters

Hyperparameters such as but not limited to learning rate, activation function and batch size determine the training behavior of a deep learning algorithm. They impact the ability of the model to learn features from an input sample and are set prior to training models. Below methods cover different hyperparameters that were used in well-known deep learning architectures for image classification.

AlexNet as compared to LeNet is not a shallow deep learning model. AlexNet consists of five convolutional layers with pooling layers after every two convolutional blocks. Additionally, there are three fully connected layers and regularization such as dropout [105]. Dropout is a simple technique that aims to reduce overfitting. It is applied in the dense layers wherein certain connections among neurons are turned off or dropped temporarily while training the model. Usual dropout rate is set to 0.5 which means that 50% of the nodes will be turned off randomly. For large dataset, Srivastava et al found default setting (drop out set to 0.5) to give optimal performance on the MNIST dataset and reduced overfitting. On the ILSVRC-2012 competition, CNN network (five convolutional layers) with dropout (0.5) achieved a top-5 error rate of 16% as compared to 26% achieved by classifier not using dropout [105]. Furthermore, AlexNet also employed using ReLU (Rectified Linear Unit) as the activation function to overcome the problem of vanishing gradient. Using ReLU [105] achieved a 20% improvement over other competitors at the ImageNet challenge. Models using the non-linear activation function ReLU, trained faster as compared to models using the tanh activation function. With using ReLU comes the need to consider the hyperparameter learning rate (affects rate of model convergence). Krizhevsky et al set the learning rate to 0.01 and then reduced it by factor of 10 manually when error did not change [105].

VGGnet is another popular architecture developed by [106] and builds on AlexNet architecture for image recognition and classification tasks. Simonyan and Zisserman have evaluated the performance of CNN with respect to increasing depth which eventually led them to first position at the 2014 ImageNet challenge. Their architecture consisted of 3x3 CNN with 11, 13 and 16 layers. Max pooling (2x2) was added after each convolutional block. Two dense layers with 4096 nodes and 1000 nodes in the last layer were used. With increase in depth of the network, there was a significant increase in the number of trainable parameters (~130M) which then led to increased computational burden [102]. A learning rate of 0.001 was used along with Glorot Uniform [107] as the random weight initializer during training. As compared to AlexNet (top-5 test error of 16%) the deeper VGGnet provided a top-5 test error of 6.8% thus showing that deeper model can lead to improved accuracy without pre-training but at the cost of increased training time.

In summary, dropout regularization (0.5) along with ReLU as the activation function and Glorot Uniform as the random initializer have been used to achieve high classification scores on image classification datasets. Additionally, initial learning rate of 0.001 provided improved model convergence on test data.

### Computational Efficiency

Dimensionality reduction refers to the technique wherein the number of samples of input data are reduced so that there occurs reduction in the number of parameters to train. As the computational requirement to train a deep learning model is proportional to the number of parameters, dimensionality reduction reduces computational burden.

The 2014-ILSVRC competition was won by the GoogleNet architecture. Their goal was to speed up training and reduce computational cost [108] and builds on top of [109] which uses the idea of having micro networks stacked together. GoogleNet uses 22 layers of CNN while reducing the trainable parameters to just 4M. As compared to previous architectures GoogleNet uses different kernel sizes for the convolution operation that can extract varying levels of features [110]. This is the underlying idea behind Inception block used by GoogleNet. Inception block serves the purpose for dimensionality reduction and hence lower the number of parameters. To achieve this, 1x1 convolution operation is utilized. Within each Inception block a 1x1 convolution is added either before the 3x3 and 5x5 convolution or after. Dropout (0.4) was used in the fully connected layers with 1024 nodes (only one dense layer). Using ensemble approach on differently tuned training models, it achieved top-5 error rate of 6%. Primary limitation of using GoogleNet is cited as the careful manual crafting of the Inception block design to suit the classification problem [111].

### Improved Performance Using Non-Complex CNN Architectures

Performance of a deep learning system refers to the model's ability to train over the input features with optimal training time and computational resources. A comparison of state-of-the-art deep CNN architectures with shallow CNN models has been provided.

A popular CNN architecture, ResNet, used 152 CNN layers for the ImageNet challenge [112]. Their architecture won the 2015-ILSVRC competition and used the idea of residual blocks. Residual blocks are also called skip connections and are based on the idea of Highway networks which aim to improve model convergence in large networks [113]. Deeper models often lead to increased training error and accuracy saturation [114], hence introduction of residual blocks which is also similar to that observed in LSTM (Long Short Term Memory) networks led to differential training based on error backpropagation [112]. The model (ResNet-101) was evaluated on the COCO object detection dataset and achieved 28% improvement than VGG-16. It should also be noted that using SimpleNet, Hasanpour et al [115] achieved 95.32% accuracy (without hyperparameter tuning and augmentation) which is similar to ResNet-110L /1202L on CIFAR-10 dataset. Their architecture, SimpleNet had 13 convolutional layers (3x3 kernel size) with max pooling (2x2). The last two layers had 1x1 convolutional operation. Since no augmentation was considered, batch normalization was used along with ReLU as the activation function. Their work hints that the optimal model performance can also be achieved with few layers in the model.

Using a balanced dataset of CT lung images, Qing et al [116] developed a CNN model with one layer convolution (7x7) and max pooling (2x2) for multi classification (5 classes) of patches of lung disease. The model had two fully connected layers with 100 and 50 nodes in the two layers leading the final layer (5 nodes). Dropout was set to 0.6 while ReLU was used as the activation function. Precision (0.80) and recall (0.76) scores show high classification scores with just one convolutional layer. The ILD dataset [117] (texture-based images) used for their model evaluation did not have increased features to learn from and thus deeper models were not required in their case.

From the above review, it is inferred that deep CNN models with different kernel size can capture both high- and low-level features from images. Deeper and shallower models were compared and was found that depending on the dataset, prior training set data augmentation should be performed to have better model convergence and classification score on the unseen test data. The next section provides details on the evaluation metrics for classification on an imbalanced imaging dataset.

### 2.5.2 Evaluation Metrics

Confusion matrix results along with classification report were used to evaluate classification results of the different models. Model accuracy nor AUC score can provide a better picture of the classifier’s performance in the case of an imbalanced multiclass dataset [36].

Keras categorical accuracy: is calculated as the percentage of predicted values that match true values and gives measure of model accuracy during training.

Precision and recall scores calculated from confusion matrix for a multi-class classification determine the true positives from all positive predictions and sensitivity respectively.

Precision: $\frac{\mathrm{TP}}{\mathrm{TP+FP}}$

It is the ratio of true positives (TP) to that of sum of true positives and false positives (FP).

Recall: $\frac{\mathrm{TP}}{\mathrm{TP+FN}}$

It is defined as the ratio of true positive to that of the sum of true positive and false negative. It is also referred as sensitivity.

F1 score: $2 \ * \left(\frac{\mathrm{Precision*Recall}}{\mathrm{Precision+Recall}}\right)$

It is a multiclass performance evaluator for each class. It is defined as twice the ratio of the product of precision and recall to sum of precision and recall.

Weighted average: $\frac{\mathrm{sum(\ (F1\ score*support\ per\ class)\ )}}{\mathrm{sum(support)}}$

It is the weighted sum of F1 scores. Support represents the number of samples of that class

Macro average: is the mean of F1 scores without weighting

Accuracy: $\frac{\mathrm{TP+TN}}{\mathrm{TP+TN+FP+FN}}$

It is the ratio of sum of true positive and true negative to that of sum of true negatives, true positives, false negatives and false positives.

Table 3: Sample confusion matrix result for multiclass predictions. Support refers to the number of samples per class.

| Predicted<br>True | Class A | Class B | Class C | Support |
|---|---|---|---|---|
| Class A | **45** | 5 | 10 | 60 |
| Class B | 10 | **35** | 5 | 50 |
| Class C | 10 | 10 | **50** | 70 |

In Table 3, precision score for class A can be calculated by taking the number of correct prediction and dividing by row sum.

$$\text{Precision (Class A)} = \frac{45}{45 + 5 + 10} = 0.75$$

Recall can be calculated by taking the number of correct predictions and dividing by the column sum in a confusion matrix.

$$\text{Recall (Class A)} = \frac{45}{45 + 10 + 10} = 0.6$$

F1 score for Class A can be calculated from recall and precision scores,

$$\text{F1 (Class A)} = 2 * \left(\frac{0.75\ *\ 0.69}{0.75 + 0.69}\right) = 0.72$$

# Chapter 3

# Methodology

We have chosen three approaches to multiclassification of liver MRI sequences

1. Baseline using imbalanced dataset,
2. Data augmentation for the minority classes and
3. Random under sampling of majority classes

Our methodology is based on the approach taken by [87], [90] and [92] which consider data level approach to deal with class imbalance in datasets. This section provides preprocessing pipeline for MRI images to be used by the deep learning model to learn features from input images. Additionally, model architectures along with hyperparameters used for training have also been provided.

## 3.1 Dataset Characteristics

In this section we have presented our imbalanced MRI sequence dataset and distribution of images per class using data augmentation and random under sampling. Furthermore, we have provided the workflow undertaken to experiment the dataset balancing approaches.

### 3.1.1 MRI Sequence Dataset

A total of 193,868 images are present in this dataset. There are 10 classes in the dataset with imbalanced distribution of images per class as observed in Figure 3. The class IPOOP contains the least number of images as compared to other classes. The number of post-contrast sequences C+ and C- are more than 20,000 each (Figure 3).

The ten classes are:

1. Apparent Diffusion Coefficient map (calculated from DWI),
2. T1-weighted post contrast imaging C-,
3. T1-weighted post contrast imaging C+,
4. T1-weighted post contrast imaging C-C+,
5. Diffusion Weighted Imaging (DWI),
6. in-phase (IP) chemical shift imaging,
7. In phase/Out of phase (IPOOP) imaging,
8. Out-phase (OOP) chemical shift imaging,

9. T2 weighted imaging and

10. Fat suppressed T2 weighted imaging (T2FS)

The purpose of these sequences is for liver observation and for determining presence/absence of LI-RADS features (major and ancillary).

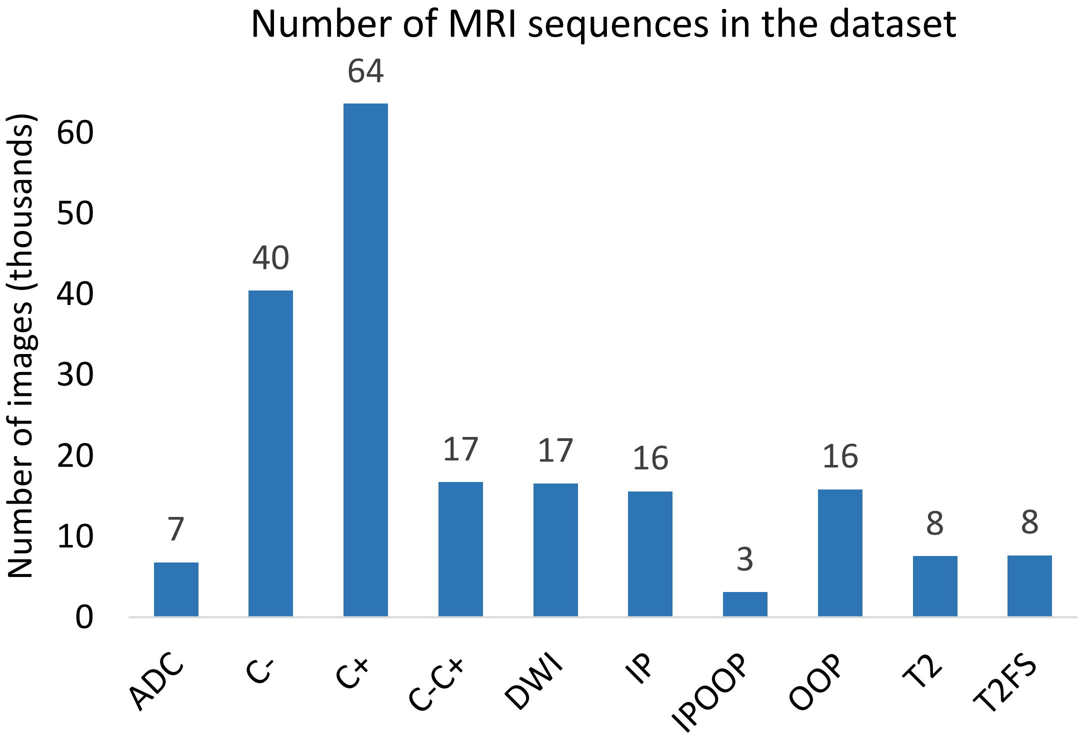


Figure 3: Distribution of images per MRI sequence. As compared to the T1-weighted post-contrast (C- and C+) phases, all other phases have significantly lower samples

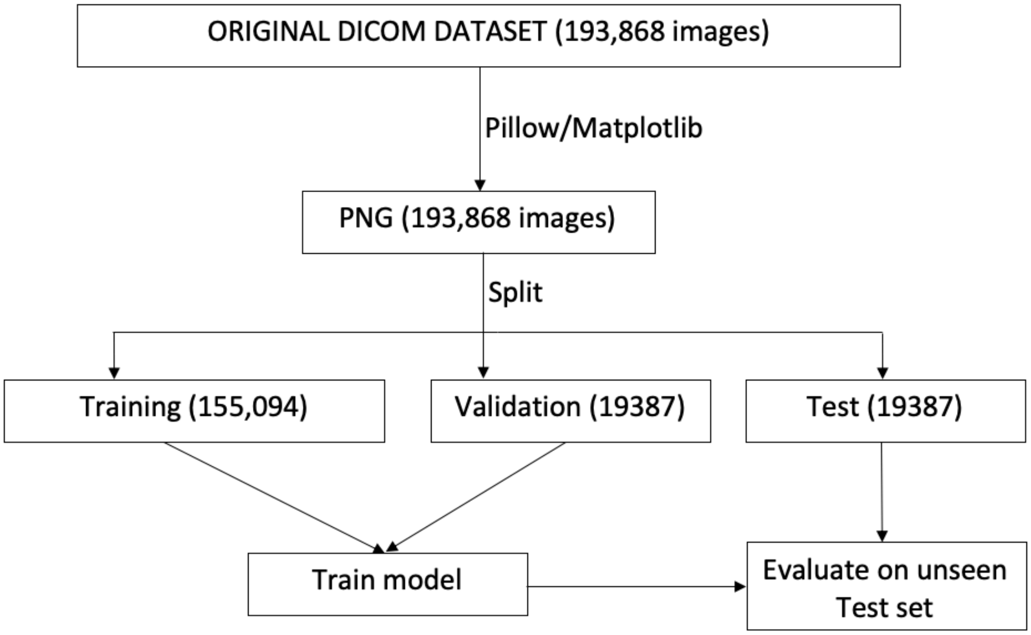


Figure 4: Dataset preparation for the MRI sequences. The dataset was randomly shuffled before partitioning data into Training, Validation, and Test sets.

The original dataset of MRI sequences with 10 classes was in the DICOM (Digital Imaging and Communication in Medicine) format (Figure 4). The MRI images were converted to PNG format using the Matplotlib library because the Keras deep learning library was unable to read the DICOM image format.

Images were randomly shuffled and then partitioned into Training (80%), Validation (10%) and Test (10%) sets. The test set is untouched during training phase wherein only the validation set is used for updating the model.

Four deep learning models (A, B, C and D) were developed and trained using two architectures (X, Y) with training and validation test from Figure 4:

Model A (baseline), architecture X, was trained using the unaugmented Training set

Model B, architecture Y, was trained using the unaugmented Training set

Model C, architecture Y, was trained using augmented Training set

Model D, architecture Y, was trained using under sampled Training set

As shown in Figure 5, the ten MRI sample images would provide a specialist with different imaging characteristics that provide information for the liver parenchyma or other features for LIRADs and detection of HCC.

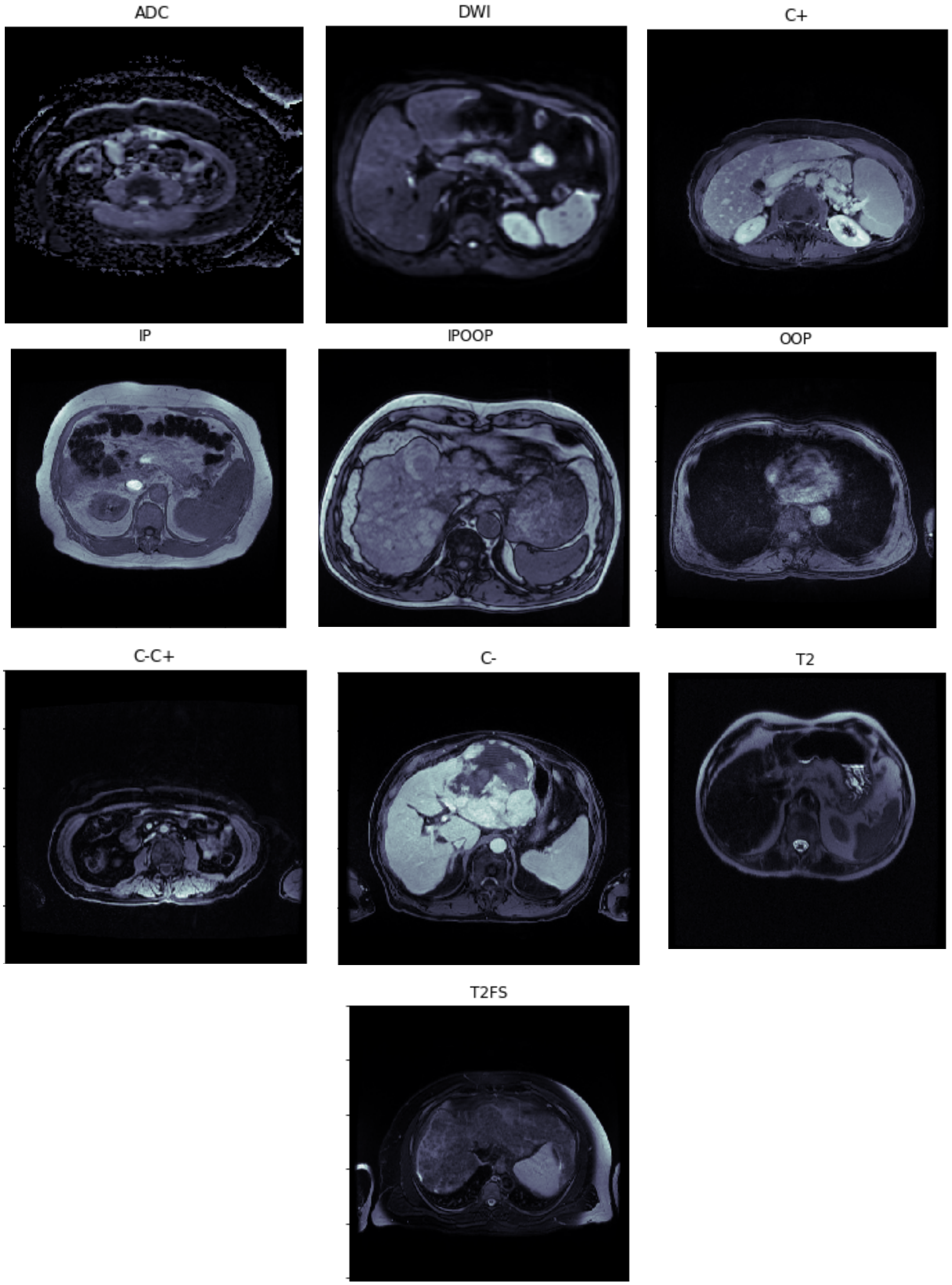


Figure 5: Sample MRI images from the sequence dataset.

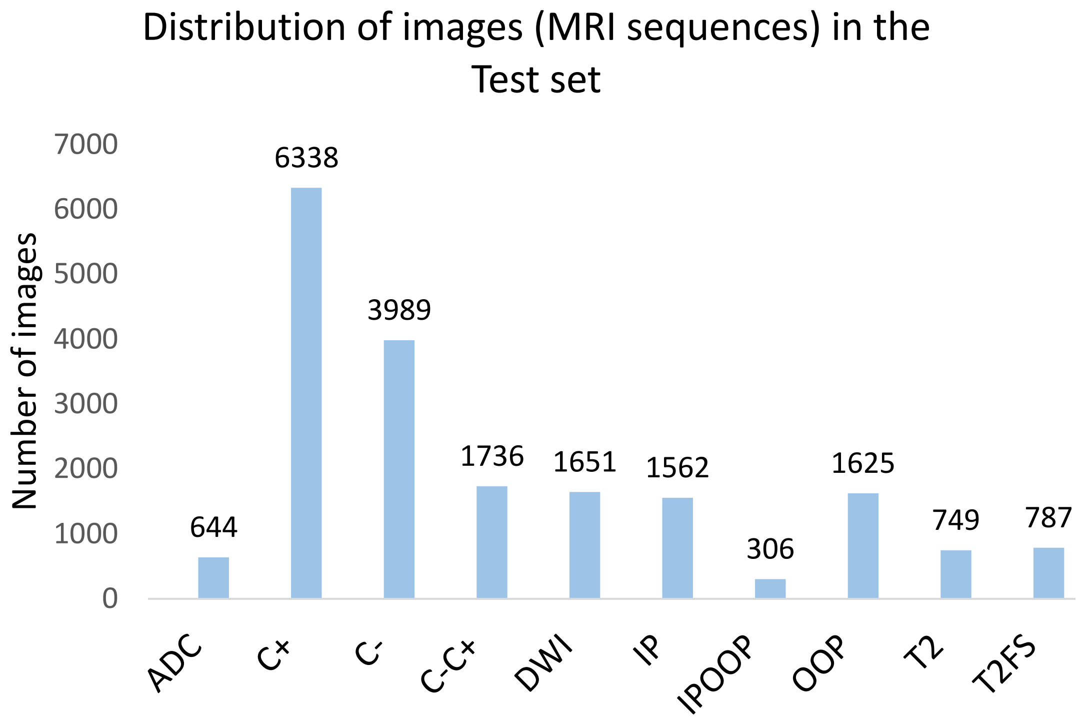


Figure 6: Test set data distribution for the MRI sequences

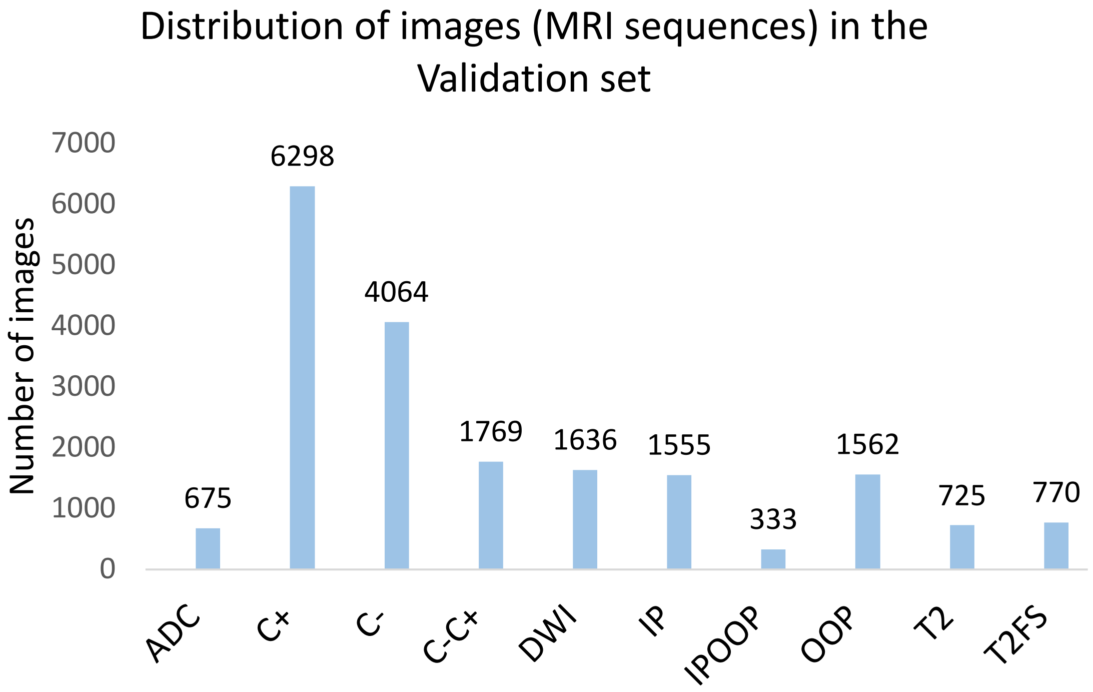


Figure 7: Validation set data distribution for the MRI sequences

From Figure 6 and Figure 7 it was observed that the distribution of images in the Test and Validation sets are not balanced; however, by visual comparison the distributions are a representative sample of the original imaging dataset by using random sampling. Both Test and Validation sets have an approximately equal total number of images per class.

## 3.2 Dataset Structuring

Since ImageDataGenerator from the Keras library was used the training, test and validation sets were organized into directory format with each class being represented as a folder as shown in below figure. The ImageDataGenerator class automatically infers the class names when images were loaded.

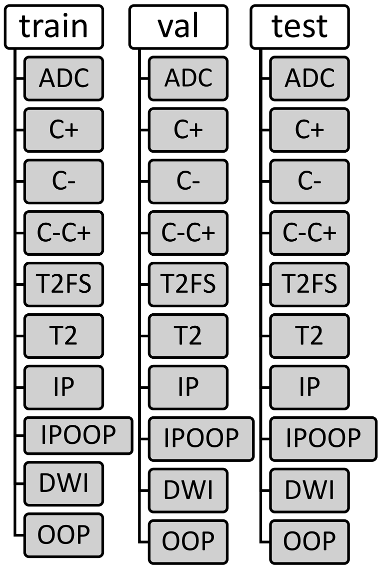


Figure 8: MRI sequence dataset organization for reading the images by ImageDataGenerator

## 3.3 Preprocessing

As observed from Figure 4, the datasets are split into Train, Validation and Test sets after shuffling the dataset. Training contains 80% of the split while validation and test sets contain 10% each. This method of dataset partitioning ensures that the test set is only used during model evaluation and validation is used during training. Furthermore, for large datasets such partitioning is preferred over cross validation due to increased number of training parameters and features in deep learning systems [118].

Augmentation was performed on only the training set after partition into the three sets which included rotation by 45 degrees, horizontal flip and vertical flip on the images (Figure 9) using the Pillow Library. This was performed to increase the balance among the classes.

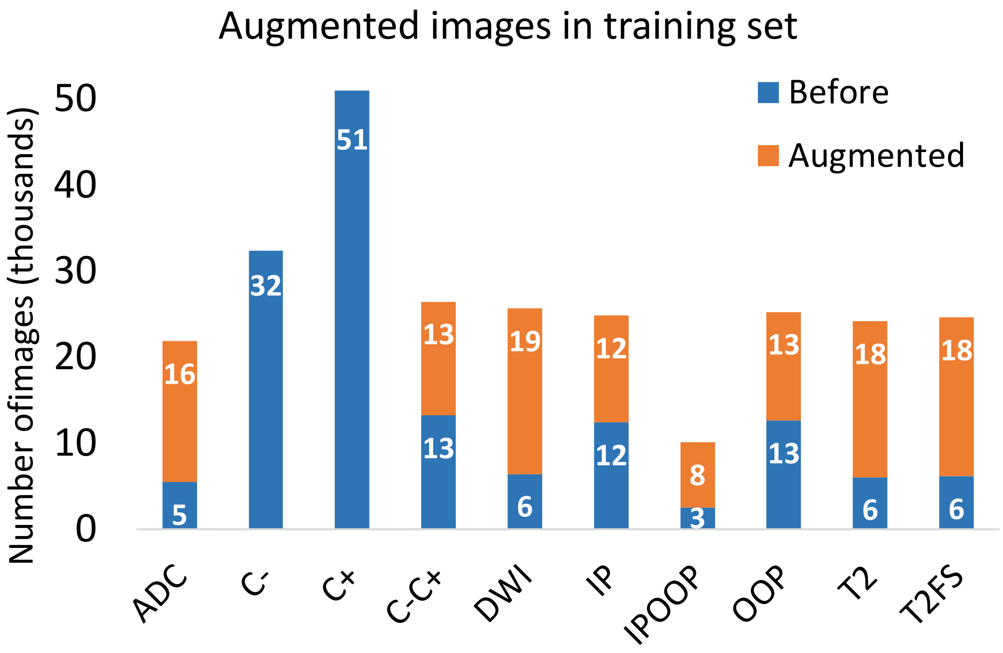


Figure 9: Increasing the number of samples per class except C- and C+ using data augmentation. Horizontal flip, vertical flip and rotation by 45 degrees was performed.

Data augmentation as observed in Figure 9 above led to artificial increase in the number of images per minority class. The augmented class included the original images plus the geometrically transformed images. As reported by [93] and [92] models trained on geometric transformations such as horizontal/vertical flip and rotation of training images had shown to provide better classification accuracy on the test set.

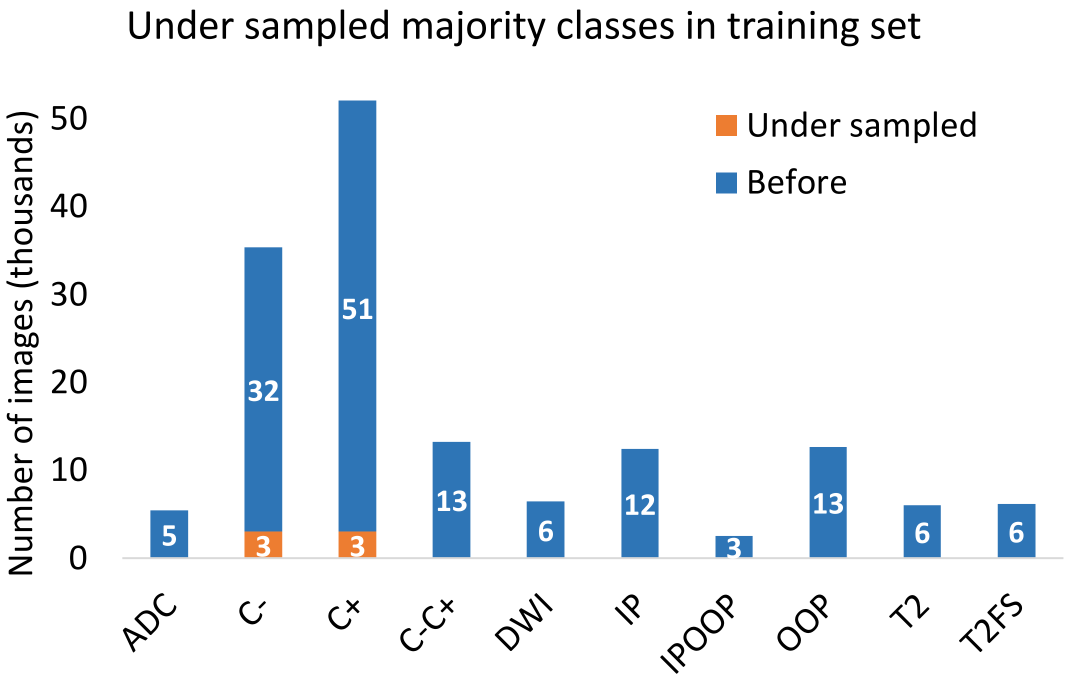


Figure 10: Random under sampling was performed for the T1-weighted post contrast (C+ and C-) classes. The number of images were under sampled to 3000 each.

Since the training of deep CNN suffers from imbalanced data distribution data augmentation is performed for the training set and model performance is compared with and without augmented training data. Additionally, model training was also performed on under sampled dataset (Figure 10).

## 3.4 Model Design and Validation

Architecture X (Figure 11) was designed for the baseline model (Model A) which only had three convolutional layers. As observed in [103], Model A was designed as a shallow model with smaller kernel or filter sizes to determine model's performance on simple architecture.

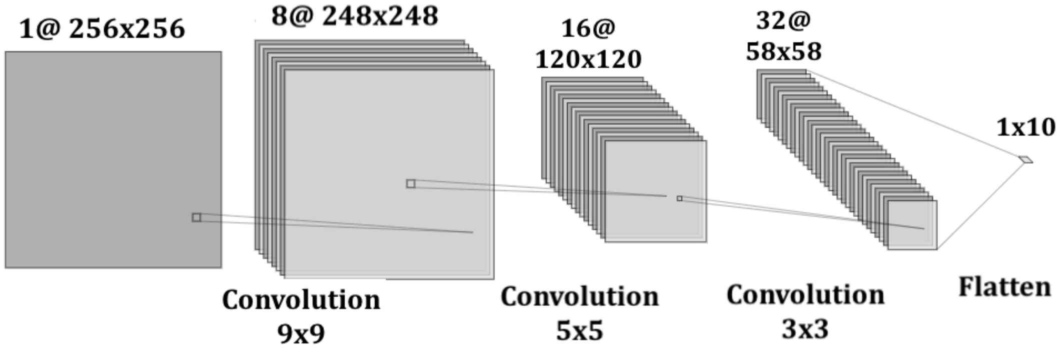


Figure 11: Architecture X for Model A (baseline model). Max pooling (not shown) was used after each convolution layer. Dense layers are not shown; however dense layers are included after the flatten layer.

Architecture Y (Figure 12) was chosen for Models B, C and D. This was designed to build on top of the Architecture X with two more convolutional layers and two additional dense layers. Additional convolutional filters would extract more higher-level features and fine-grained features that could increase model's performance [104].

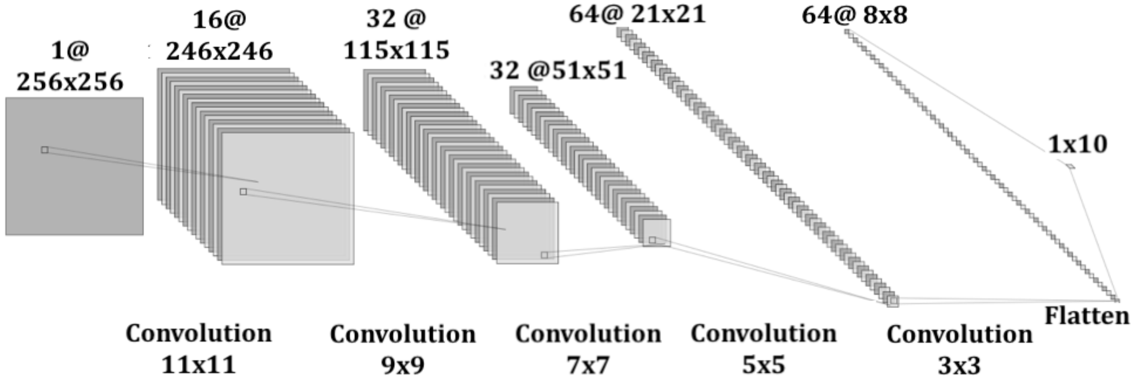


Figure 12: Architecture Y for models B, C and D. The above shows the CNN layers. Dense layers and max pooling are not shown; however, dense layers are included after the flatten layer.

For the sequence classification (10 classes), two model architectures were constructed. The baseline model (Model A) had 3 convolutional layers and one dense layer (1028 nodes) while a deeper network (Model B, C and D) with 5 convolutional layers and three dense layers (1028 nodes each) was also devised. A large kernel size of 9x9 was used in Model A which aims to capture high level features of the input image while the smaller kernel sizes (5x5 and 3x3) in the subsequent layers look for fine grained features [59].

Models were trained on both architectures. Model A and Model B were trained on imbalanced training set (Figure 3) while Model C was trained on augmented training set (Figure 9) and Model D was trained on the under sampled training set (Figure 10). Similar scheme proposed by [88][89] (refer to 2.4.1) was followed to compare model performance when image augmentation and under sampling was used. Since the geometric transformations do not alter the most of property related to the MRI sequence images, training CNN models with such image augmentation can provide robustness to image spatial orientations. To better assess the classification scores of models, F1 score (a multiclass evaluator) metric was used to compare among the different models.

Furthermore, to investigate the effect of adding more layers on the classification accuracy, Architecture Y was used as it was observed by Szegedy that adding more layers increases model convergence [110]. Model B, C and D had the same architecture.

### 3.4.1 Model Hyperparameters

Keras' Glorot Uniform was used as the random weight initializer with a learning rate of 0.001 as reported by [94][102]. Learning rate refers to the ability of the model to converge to a classification problem. High learning rates will lead to less number of epochs required but most often result in sub-optimal accuracy [105]. Batch size of 64 was used for model training. The hyperparameter, batch size, refers to the number of samples or images that will be passed to the model during training before updating the model weights. A batch size of 64 in our big dataset ensures optimal training efficiency and model convergence.

Adam optimizer was used as it has been reported to provide faster model convergence [65][93]. ReLU was used as the activation function along with dropout (0.5) regularization [97][103][92]. Dropout regularization was used after each dense layer for Models B, C and D but was not used in the baseline model (Model A).

Table 4: Hyperparameters used for training the models on Architectures X and Y. A batch size of 64 was used for all models.

| **Model** | **Hyperparameters** | | | |
|---|---|---|---|---|
| | **Learning rate** | **Activation function** | **Optimizer** | **Early Stopping** |
| Model A | 0.001 | ReLU | Adam | Patience value of 7 |
| Model B | 0.001 | ReLU | Adam | Patience value of 7 |
| Model C | 0.001 | ReLU | Adam | Patience value of 7 |
| Model D | 0.001 | ReLU | Adam | Patience value of 7 |

### 3.4.2 Development Environment

Development environment used for training was TensorFlow 2.0, Python 3.8.2 and Lambda Labs Stack (3x2080Ti GPU). Libraries used for the model training and evaluation were Keras and ScikitLearn. ImageDataGenerator class was used for loading the datasets. Training was run for maximum of 12 epochs for each model.

# Chapter 4

# Results

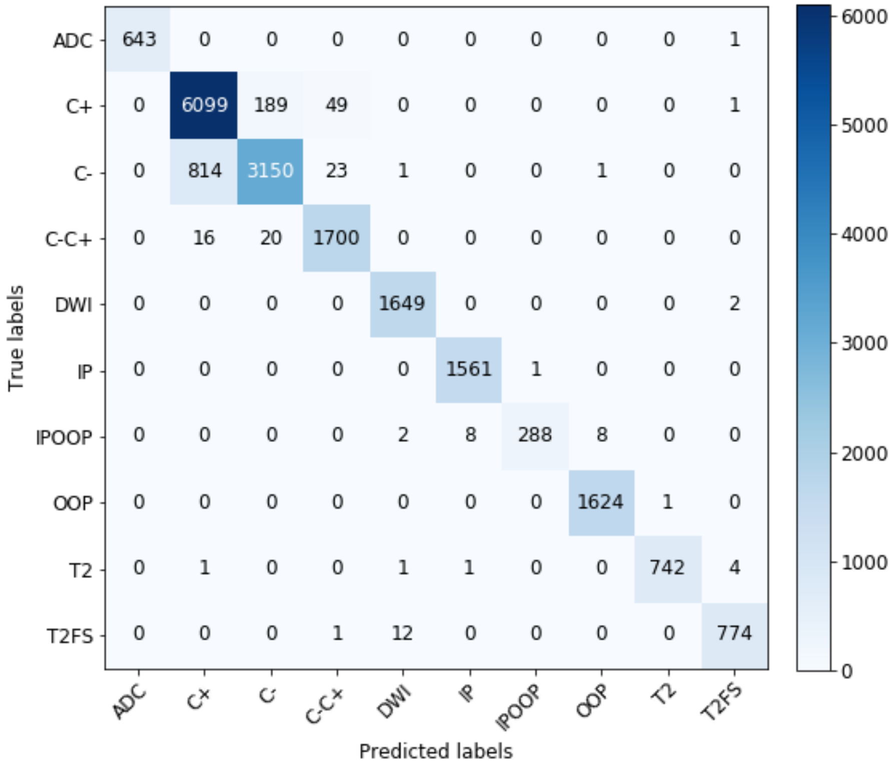
In this section, results of the four trained models are presented. Models A, B, C and D were evaluated for their predictive ability to classify MRI sequences from the unseen test set images (19,387).

Multi-class F1-scores for each class label prediction and confusion matrix results were reported. Confusion matrix in particular provides a more robust measure of sensitivity and precision scores for the model performance for imbalanced data distribution. Furthermore, classification reports for each model's performance on classifying the different classes was also included.

## 4.1 Model A - Classification Results

We report precision, recall and F1 scores calculated from confusion matrix (Figure 13) for baseline model (Model A) from predictions on test set. Classification scores are reported in

Table 5.



| True labels \ Predicted labels | ADC | C+ | C- | C-C+ | DWI | IP | IPOOP | OOP | T2 | T2FS |
|---|---|---|---|---|---|---|---|---|---|---|
| ADC | 643 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |
| C+ | 0 | 6099 | 189 | 49 | 0 | 0 | 0 | 0 | 0 | 1 |
| C- | 0 | 814 | 3150 | 23 | 1 | 0 | 0 | 1 | 0 | 0 |
| C-C+ | 0 | 16 | 20 | 1700 | 0 | 0 | 0 | 0 | 0 | 0 |
| DWI | 0 | 0 | 0 | 0 | 1649 | 0 | 0 | 0 | 0 | 2 |
| IP | 0 | 0 | 0 | 0 | 0 | 1561 | 1 | 0 | 0 | 0 |
| IPOOP | 0 | 0 | 0 | 0 | 2 | 8 | 288 | 8 | 0 | 0 |
| OOP | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1624 | 1 | 0 |
| T2 | 0 | 1 | 0 | 0 | 1 | 1 | 0 | 0 | 742 | 4 |
| T2FS | 0 | 0 | 0 | 1 | 12 | 0 | 0 | 0 | 0 | 774 |

Figure 13: Confusion matrix results for Model A (sequence Test dataset)

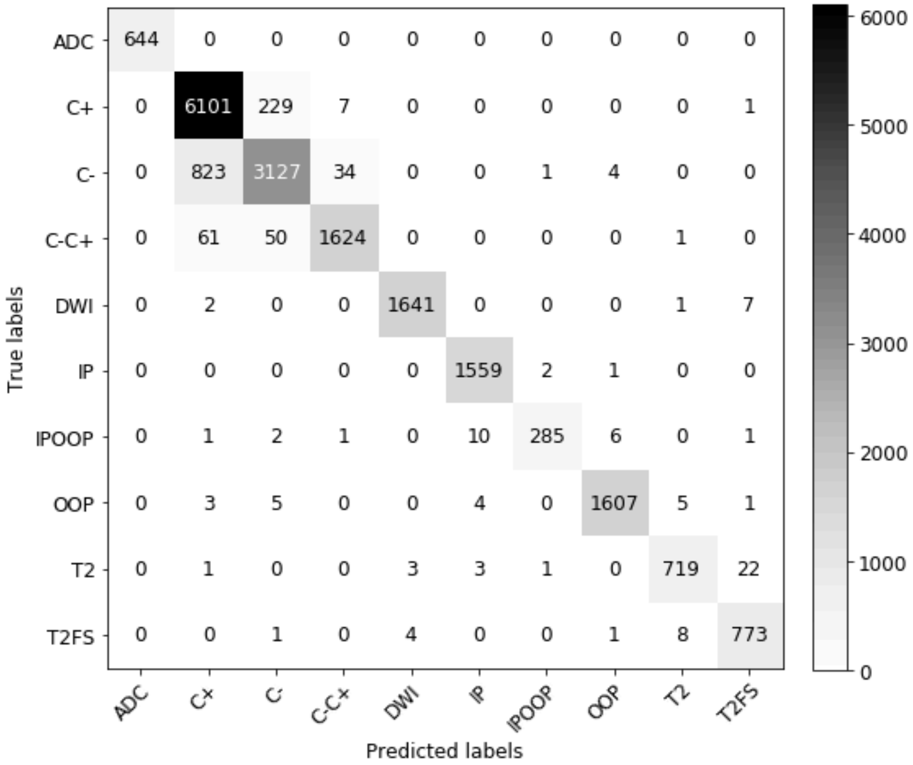
Table 5: Classification report for (Model A) with sequence Test dataset

| | precision | recall | F1-score | support |
|---|---|---|---|---|
| ADC | 1 | 1 | 1 | 644 |
| C+ | 0.88 | 0.96 | 0.92 | 6338 |

| | | | | |
|---|---|---|---|---|
| C- | 0.94 | 0.79 | 0.86 | 3989 |
| C-C+ | 0.96 | 0.98 | 0.97 | 1736 |
| DWI | 0.99 | 1 | 0.99 | 1651 |
| IP | 0.99 | 1 | 1 | 1562 |
| IPOOP | 1 | 0.94 | 0.97 | 306 |
| OOP | 0.99 | 1 | 1 | 1625 |
| T2 | 1 | 0.99 | 0.99 | 749 |
| T2FS | 0.99 | 0.98 | 0.99 | 787 |
| accuracy | | | 0.94 | 19387 |
| macro avg | 0.97 | 0.96 | 0.97 | 19387 |
| weighted avg | 0.94 | 0.94 | 0.94 | 19387 |

## 4.2 Model B - Classification Results

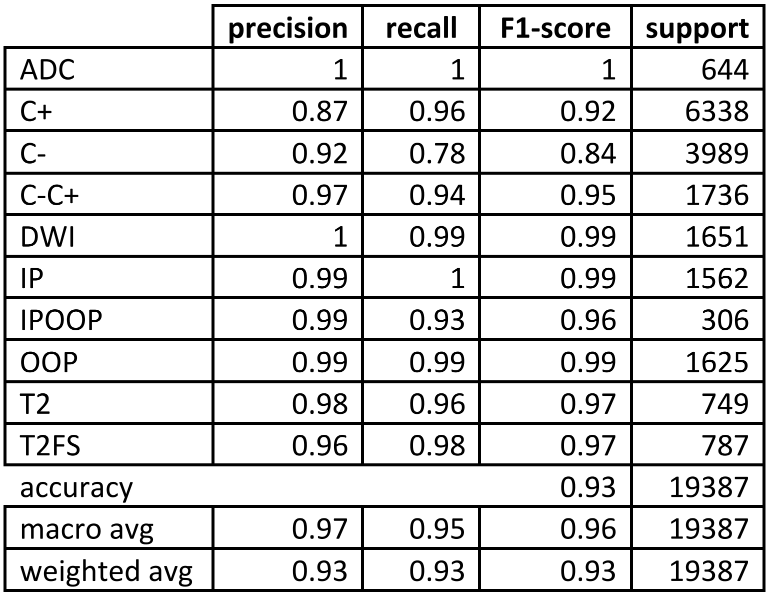
Model B was trained on the Architecture Y and we report the classification scores in Table 6.



| | ADC | C+ | C- | C-C+ | DWI | IP | IPOOP | OOP | T2 | T2FS |
|---|---|---|---|---|---|---|---|---|---|---|
| ADC | 644 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| C+ | 0 | 6101 | 229 | 7 | 0 | 0 | 0 | 0 | 0 | 1 |
| C- | 0 | 823 | 3127 | 34 | 0 | 0 | 1 | 4 | 0 | 0 |
| C-C+ | 0 | 61 | 50 | 1624 | 0 | 0 | 0 | 0 | 1 | 0 |
| DWI | 0 | 2 | 0 | 0 | 1641 | 0 | 0 | 0 | 1 | 7 |
| IP | 0 | 0 | 0 | 0 | 0 | 1559 | 2 | 1 | 0 | 0 |
| IPOOP | 0 | 1 | 2 | 1 | 0 | 10 | 285 | 6 | 0 | 1 |
| OOP | 0 | 3 | 5 | 0 | 0 | 4 | 0 | 1607 | 5 | 1 |
| T2 | 0 | 1 | 0 | 0 | 3 | 3 | 1 | 0 | 719 | 22 |
| T2FS | 0 | 0 | 1 | 0 | 4 | 0 | 0 | 1 | 8 | 773 |

Figure 14: Confusion matrix result for Model B (sequence Test dataset)

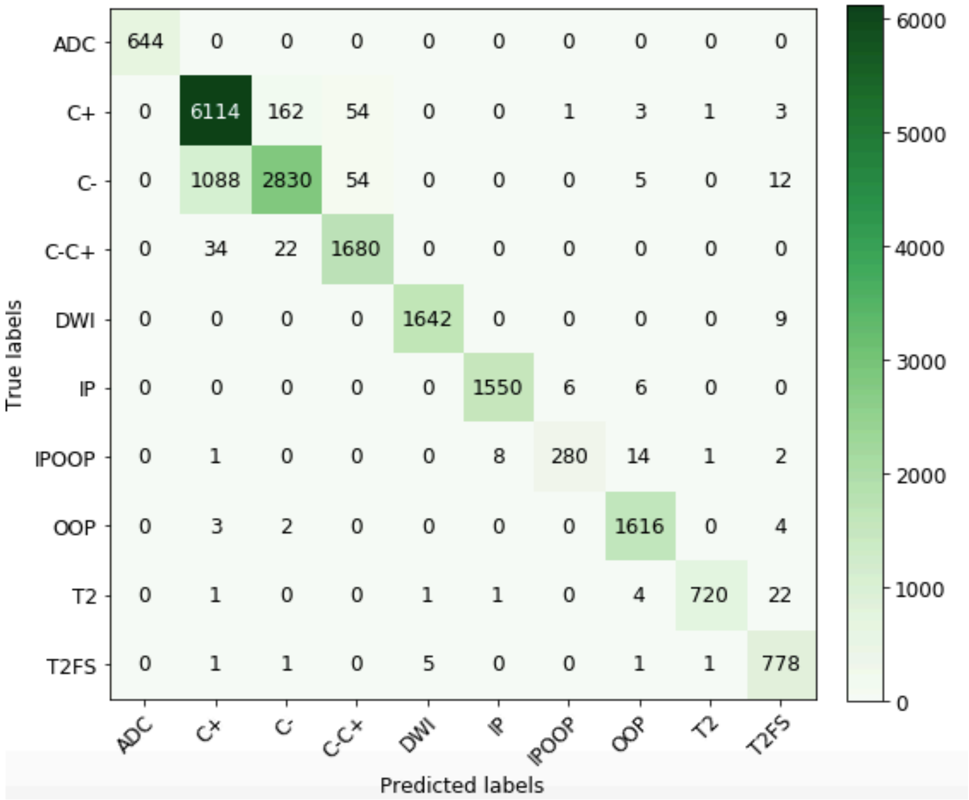
Table 6: Classification report for Model B with sequence Test dataset



| | precision | recall | F1-score | support |
|---|---|---|---|---|
| ADC | 1 | 1 | 1 | 644 |
| C+ | 0.87 | 0.96 | 0.92 | 6338 |
| C- | 0.92 | 0.78 | 0.84 | 3989 |
| C-C+ | 0.97 | 0.94 | 0.95 | 1736 |
| DWI | 1 | 0.99 | 0.99 | 1651 |
| IP | 0.99 | 1 | 0.99 | 1562 |
| IPOOP | 0.99 | 0.93 | 0.96 | 306 |
| OOP | 0.99 | 0.99 | 0.99 | 1625 |
| T2 | 0.98 | 0.96 | 0.97 | 749 |
| T2FS | 0.96 | 0.98 | 0.97 | 787 |
| accuracy | | | 0.93 | 19387 |
| macro avg | 0.97 | 0.95 | 0.96 | 19387 |
| weighted avg | 0.93 | 0.93 | 0.93 | 19387 |

## 4.3 Model C - Classification Results

Using Architecture Y, the classification predictions on test set for Model C are reported here. Model C was trained on the augmented training set. Model B and C only differ on the type of training data that was used to train the models. Results from Figure 15 and Table 7 show similar precision and recall scores as compared to Model B.



| | ADC | C+ | C- | C-C+ | DWI | IP | IPOOP | OOP | T2 | T2FS |
|---|---|---|---|---|---|---|---|---|---|---|
| ADC | 644 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| C+ | 0 | 6114 | 162 | 54 | 0 | 0 | 1 | 3 | 1 | 3 |
| C- | 0 | 1088 | 2830 | 54 | 0 | 0 | 0 | 5 | 0 | 12 |
| C-C+ | 0 | 34 | 22 | 1680 | 0 | 0 | 0 | 0 | 0 | 0 |
| DWI | 0 | 0 | 0 | 0 | 1642 | 0 | 0 | 0 | 0 | 9 |
| IP | 0 | 0 | 0 | 0 | 0 | 1550 | 6 | 6 | 0 | 0 |
| IPOOP | 0 | 1 | 0 | 0 | 0 | 8 | 280 | 14 | 1 | 2 |
| OOP | 0 | 3 | 2 | 0 | 0 | 0 | 0 | 1616 | 0 | 4 |
| T2 | 0 | 1 | 0 | 0 | 1 | 1 | 0 | 4 | 720 | 22 |
| T2FS | 0 | 1 | 1 | 0 | 5 | 0 | 0 | 1 | 1 | 778 |

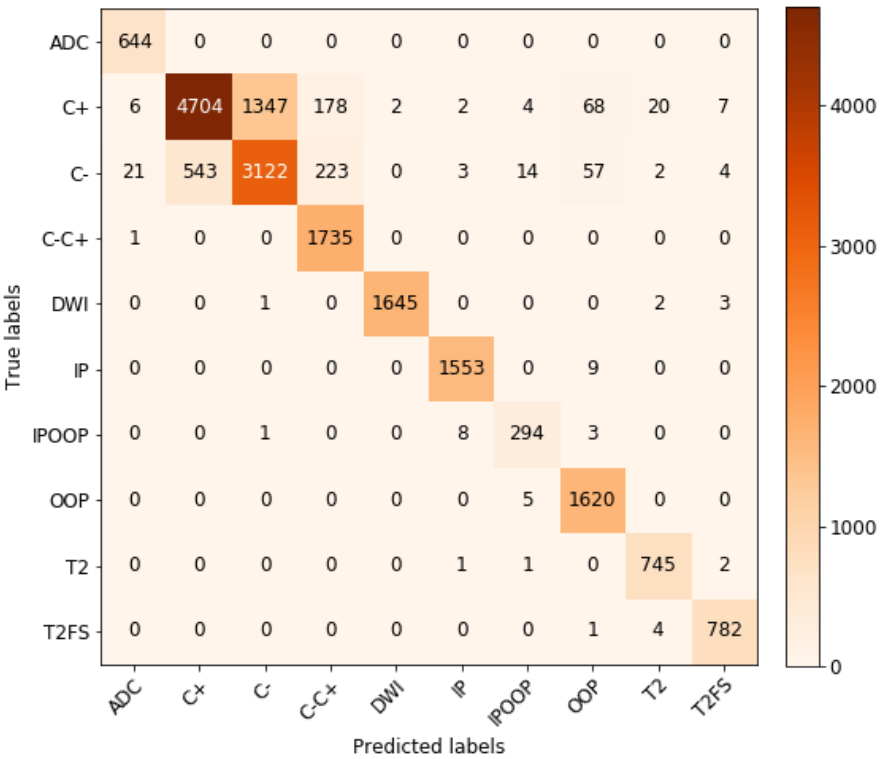
Figure 15: Confusion matrix results for Model C (augmented Training set)

Table 7: Classification report for Model C (augmented Training set)

| | **precision** | **recall** | **F1-score** | **support** |
|---|---|---|---|---|
| ADC | 1 | 1 | 1 | 644 |
| C+ | 0.84 | 0.96 | 0.9 | 6338 |
| C- | 0.94 | 0.71 | 0.81 | 3989 |
| C-C+ | 0.94 | 0.97 | 0.95 | 1736 |
| DWI | 1 | 0.99 | 1 | 1651 |
| IP | 0.99 | 0.99 | 0.99 | 1562 |
| IPOOP | 0.98 | 0.92 | 0.94 | 306 |
| OOP | 0.98 | 0.99 | 0.99 | 1625 |
| T2 | 1 | 0.96 | 0.98 | 749 |
| T2FS | 0.94 | 0.99 | 0.96 | 787 |
| accuracy | | | 0.92 | 19387 |
| macro avg | 0.96 | 0.95 | 0.95 | 19387 |
| weighted avg | 0.93 | 0.92 | 0.92 | 19387 |

## 4.4 Model D - Classification Results

Classification predictions for Model D which was trained on the under sampled training set has been reported here. The majority class (T1-weighted post contrast C+ and C-) were under sampled to reduce class imbalance.

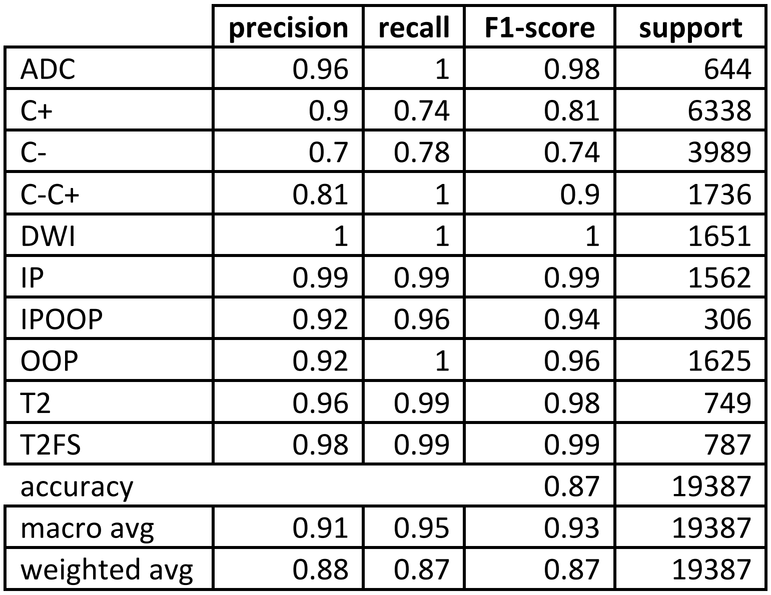


| True labels \ Predicted labels | ADC | C+ | C- | C-C+ | DWI | IP | IPOOP | OOP | T2 | T2FS |
|---|---|---|---|---|---|---|---|---|---|---|
| ADC | 644 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| C+ | 6 | 4704 | 1347 | 178 | 2 | 2 | 4 | 68 | 20 | 7 |
| C- | 21 | 543 | 3122 | 223 | 0 | 3 | 14 | 57 | 2 | 4 |
| C-C+ | 1 | 0 | 0 | 1735 | 0 | 0 | 0 | 0 | 0 | 0 |
| DWI | 0 | 0 | 1 | 0 | 1645 | 0 | 0 | 0 | 2 | 3 |
| IP | 0 | 0 | 0 | 0 | 0 | 1553 | 0 | 9 | 0 | 0 |
| IPOOP | 0 | 0 | 1 | 0 | 0 | 8 | 294 | 3 | 0 | 0 |
| OOP | 0 | 0 | 0 | 0 | 0 | 0 | 5 | 1620 | 0 | 0 |
| T2 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 745 | 2 |
| T2FS | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 4 | 782 |

Figure 16: Confusion matrix results for Model D (under sampled Training set)

From Figure 16, it was observed that Model D showed overall good prediction for the minority classes while classification prediction was lower for the T1-weighted post contrast C+ and C- classes.

Table 8: Classification report for Model D (under sampled Training set)



| | **precision** | **recall** | **F1-score** | **support** |
|---|---|---|---|---|
| ADC | 0.96 | 1 | 0.98 | 644 |
| C+ | 0.9 | 0.74 | 0.81 | 6338 |
| C- | 0.7 | 0.78 | 0.74 | 3989 |
| C-C+ | 0.81 | 1 | 0.9 | 1736 |
| DWI | 1 | 1 | 1 | 1651 |
| IP | 0.99 | 0.99 | 0.99 | 1562 |
| IPOOP | 0.92 | 0.96 | 0.94 | 306 |
| OOP | 0.92 | 1 | 0.96 | 1625 |
| T2 | 0.96 | 0.99 | 0.98 | 749 |
| T2FS | 0.98 | 0.99 | 0.99 | 787 |
| accuracy | | | 0.87 | 19387 |
| macro avg | 0.91 | 0.95 | 0.93 | 19387 |
| weighted avg | 0.88 | 0.87 | 0.87 | 19387 |

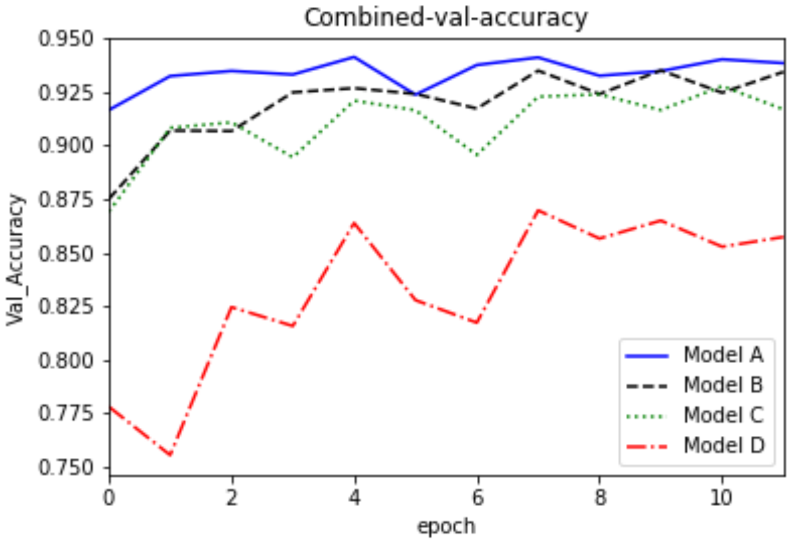


Figure 17: Validation accuracy for all models during training. Model D shows lowest accuracy

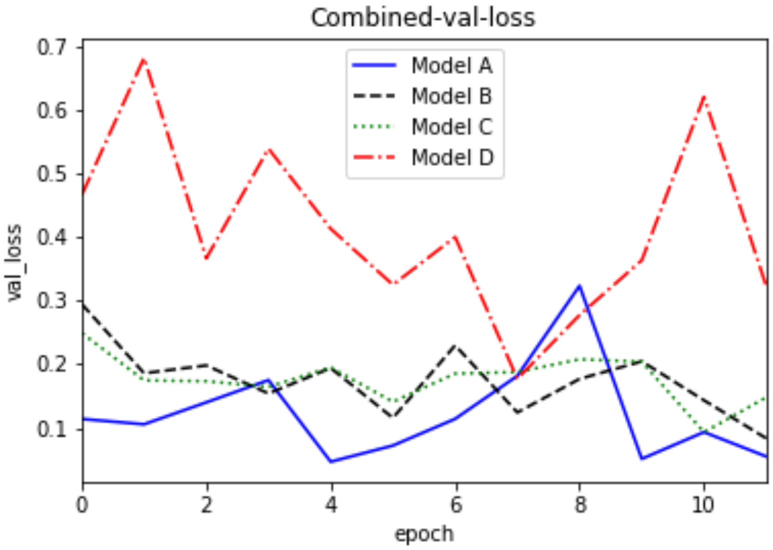
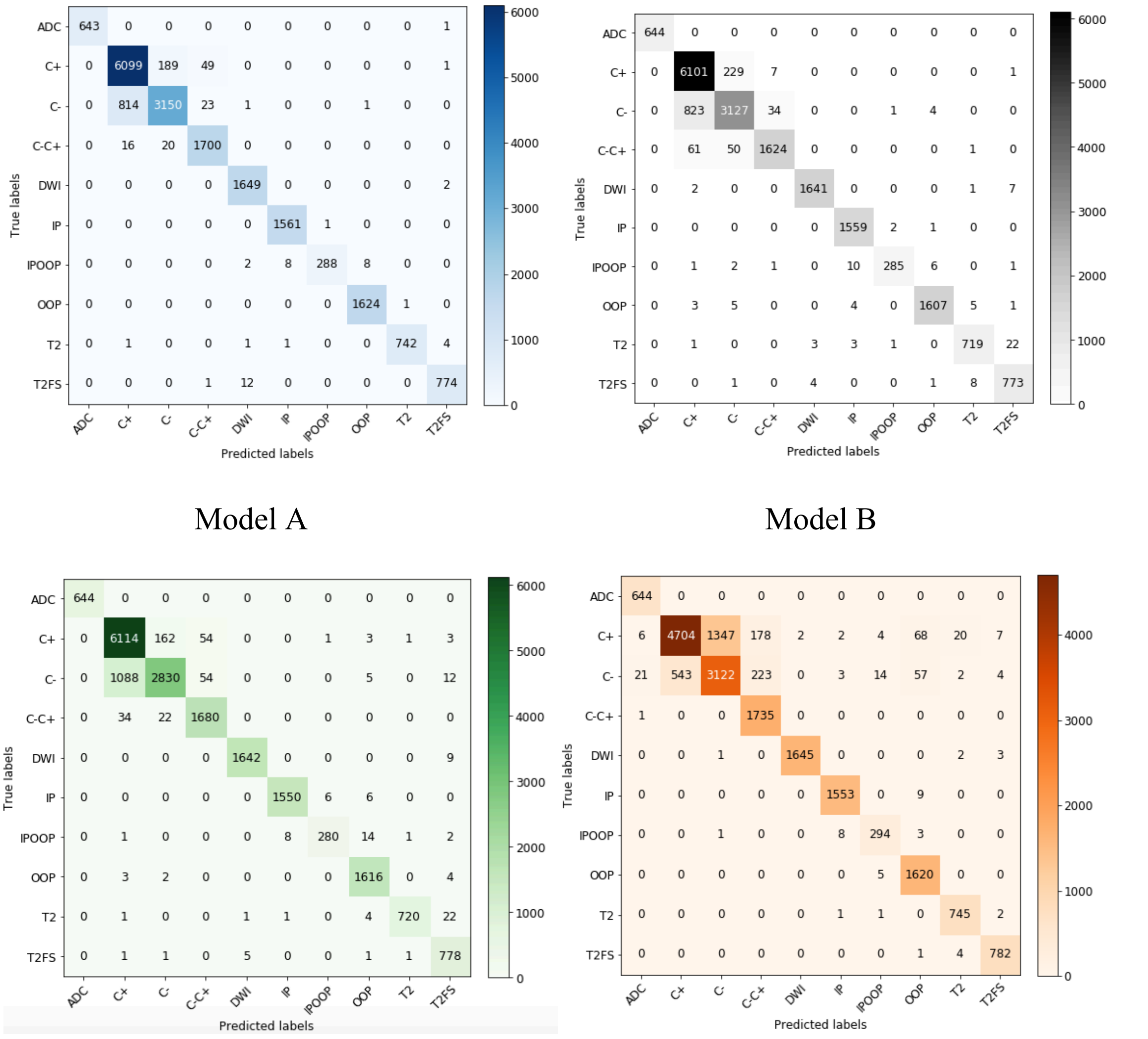


Figure 18: Validation loss for all models during training.



Model A

Model B

Model C

Model D

Figure 19: Combined visualization of confusion matrix results for Models A, B, C and D

Comparing confusion matrix results from all four models in Figure 19, it was observed that Model D showed least correct predictions (4704) for the majority C+ class. There were more incorrect C+ predictions for Model D (1347) than Model C (162) while minority class predictions were similar to that of Model C.

Figure 17 shows that the three models (Models A, B and C) show high accuracy (more than 85%) during training using the validation set. Figure 18 reports that the models show low validation error after the fourth epoch. The validation loss decreases as the validation accuracy increases and shows that the models converge to the validation set in just few epoch runs. Furthermore, a steady decrease in validation loss was not observed from Figure 18 for Models B and C but they had lowered validation loss as compared to the baseline model (Model A) after the fifth epoch.

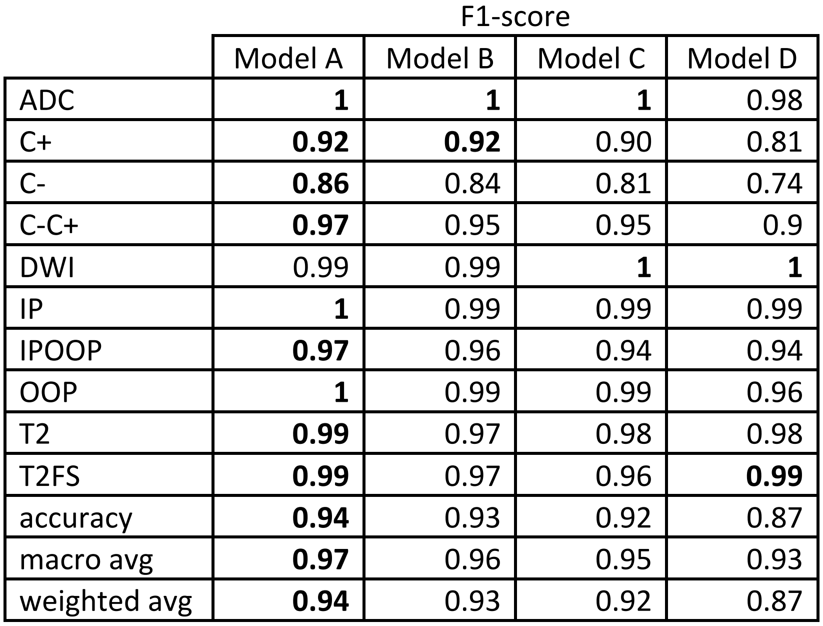
From Figure 13, Figure 14 and Figure 15 confusion matrix results for Models A, B and C it is reported that the classification scores of individual classes on the test set images is observed to have a bias to the majority class. However, the model predicts the minority test set images with high accuracy with all three models (Model A, B and C). Model C which used the augmented training set further shows that the model still has a bias to the majority class and there are 1088 incorrect predictions to the T1-weighted post contrast C- sequence.

The F1-score gives a measure of the multiclass prediction accuracy and for the minority classes from Table 5, Table 6 and Table 7 it is observed that as compared to the majority C+ and C- MRI sequences, high precision and recall scores were observed. Recall scores are representative of sensitivity of the model to class predictions and high recall scores from Table 5, Table 6 and Table 7 indicate that the model predictions return most of the relevant class predictions. For ADC (Apparent Diffusion Coefficient) class in Table 5, Table 6 and Table 7, a recall score of 1 indicates that all images of that particular class were labelled as ADC by the model however, it does not provide information for the number of images that were incorrectly labelled as ADC from the other classes. Similarly, a precision score of 1 for ADC indicates that every image labelled by the model as ADC does indeed belong to the ADC class.

Confusion matrix results from Figure 16 and classification scores reported in Table 8 show that Model D has poor accuracy and F1 score as compared to Model C for majority class predictions. Furthermore, from Figure 17 and Figure 18 it was also observed that Model D had comparatively high validation loss to that of Models A, B and C.

## 4.5 Summary

Table 9: F1 score report summary per class for all models trained on the MRI sequence datasets.



F1-score

| | Model A | Model B | Model C | Model D |
|---|---|---|---|---|
| ADC | **1** | **1** | **1** | 0.98 |
| C+ | **0.92** | **0.92** | 0.90 | 0.81 |
| C- | **0.86** | 0.84 | 0.81 | 0.74 |
| C-C+ | **0.97** | 0.95 | 0.95 | 0.9 |
| DWI | 0.99 | 0.99 | **1** | **1** |
| IP | **1** | 0.99 | 0.99 | 0.99 |
| IPOOP | **0.97** | 0.96 | 0.94 | 0.94 |
| OOP | **1** | 0.99 | 0.99 | 0.96 |
| T2 | **0.99** | 0.97 | 0.98 | 0.98 |
| T2FS | **0.99** | 0.97 | 0.96 | **0.99** |
| accuracy | **0.94** | 0.93 | 0.92 | 0.87 |
| macro avg | **0.97** | 0.96 | 0.95 | 0.93 |
| weighted avg | **0.94** | 0.93 | 0.92 | 0.87 |

As shown in Table 9, the models show high F1 scores indicating that the misclassifications are very low. Macro average score represents the average score for all classes while the weighted average considers the relative number of samples per class (column named support). From Table 9, all models have a macro / weighted average score of more than 0.9 and show high accuracy. It was also observed that Model C had a slightly lower accuracy (0.92) as compared to Model A (0.94) while Model D had the lowest accuracy (0.87). Lastly, Model D reported F1 score of 0.74 and 0.81for the C+ and C- class.

# Chapter 5

# Discussion

## 5.1 Results Analysis

From the experiments performed. It was expected that the model trained on the augmented training set would show higher classification scores as compared to the baseline model that was trained on the imbalanced MRI sequence dataset. As [71] and [72] have shown that oversampling minority class using augmentation to balance the dataset reduces model bias to majority class, it was expected that Model A would have poor classification performance on the test set. However, results from Table 5 indicate that Model A did not suffer from imbalance in the dataset.

Comparing classification results for all models (Model A, B, C and D) it was found that the baseline model (Model A) provided an accuracy of 94% with high F1 scores for all classes. Model B had two additional dense layers and five convolutional layers as compared to the baseline model. Adding more convolutional layers (Model B) increases the number of trainable parameters and it could lead to overfitting wherein the model over fits on the training dataset [110]. Results from table 6 show that Model B had comparatively lower F1 score to that of Model A however, Figure 17 and Figure 18 do not show signs of overfitting. Furthermore, dropout regularization (0.5) was used which prevents overfitting.

A potential cause of Model B's performance over Model A could also be due to a larger kernel size of 11x11 that was used in the Architecture Y. Larger kernel size could have affected the model ability to generalize over the test set. Ozturk et al have experimented different convolutional kernel sizes on histopathology image set and report that large kernel sizes could lead to higher validation loss [119] as was observed in Figure 18 for Model B.

Buda et [72] have reported that oversampling on the training data prior to training the CNN models provide better model convergence and reduces bias to majority class. They had used both CIFAR-10 and MNIST dataset and found that with the increased number of minority classes (eight classes) along with increase in imbalance ratio from 100, model classification suffers. However, our results show that Models A, B and C show similar classification performance and that Model A has the best performance.

Furthermore, model trained using the random under sampled training set showed poor classification scores as compared to Models A, B and C. It was expected that under sampling of the majority class would reduce the classification performance for the overrepresented class. This was supported by the results from Table 9. A F1 score of 0.74 was observed for C- class while Models A, B and C showed a score of more than 0.8. The F1 score for the minority class was not affected and was similar to the baseline model (Model A). Removal of samples from the dataset to balance it prior to training has shown to affect the overall model performance as was reported by [12].

From Figure 17 and Figure 18 it was also observed that compared to data augmentation method, random under sampling provided lower validation accuracy and higher validation loss. After the third epoch Model C shows an accuracy of 90% while Model D showed less than 85% accuracy. Additionally, Figure 18 also shows that validation loss for Model D is highest among all the models trained. Our results further support that under sampling approach affects the model's ability to generalize on the unseen test set images.

Furthermore, overfitting for Models A, B, C and D was not observed as confusion matrix results from Figure 19 show that the classification predictions on the test were correctly predicted for minority and majority classes. For the majority class (T1-weighted post contrast C+ and C-) there were some misclassifications such as 814 images were incorrectly predicted to C- however, most were predicted correctly. Additionally, no increase in validation loss (Figure 18) observed during model training which further suggests that overfitting was not observed [104]. Lastly, drop out regularization was used during training which has shown to reduce risk of overfitting [105].

A primary limitation of Model D was the random deletion of the majority class images which could have affected classification scores. Future extension would look at addressing the under-sampling approach using expert intervention since some of the T1-weighted post contrast sequences provide important information for HCC features. Another limitation of our approach was the exploration of different optimizer such as AdaGrad and the use of Leaky ReLU as the activation function to compare it with Adam optimizer and ReLU respectively for imbalanced datasets. Furthermore, model performance using larger batch size than 64 was not evaluated due to memory limitation of the GPU. As larger batch would reduce the training time, future work would include a learning rate scheduler and using batch size of 128 or 256 to evaluate optimal performance for big datasets.

Lastly, another limitation of model design was that the architecture depth as variable beyond Architecture Y was not examined.

# Conclusion

In conclusion, we have developed and tested four deep CNN classifiers to classify MRI sequence from an imbalanced distribution in big datasets. Our objective was to balance dataset prior to training using data augmentation and random under sampling and evaluate the performance of deep CNN models when trained using imbalanced data distribution, augmented balanced distribution of training set and lastly under sampled distribution of majority classes.

Imbalance in datasets had shown to affect model performance and we had performed data augmentation (horizontal flip, vertical flip and 45-degree rotation) to increase the balance among classes. Random under sampling was also performed prior to training which removed samples from the T1-weighted post contrast C+ and C- class and was compared with the augmentation approach.

We conclude that model performance did not suffer when they were trained using imbalanced distribution of images among classes. Model A gave a macro average F1 score of 0.97 (Table 9). Comparatively, Model D showed poor classification score (macro average F1 score of 0.93) when the majority classes were randomly under sampled. Lastly, our simple CNN network architecture (Architecture X) with only three convolutional layers and one dense layer provided the best classification prediction over the models trained using Architecture Y (Figure 12). However, future experiments should be performed with an extension to Architecture Y wherein deeper architectures would be designed to evaluate for architecture depth as a variable for model performance.

Future work will explore the use of our CNN classifiers in cases of small multi-class datasets with high class imbalance. Another future implementation of our approach will be towards real-time classification of medical images. Furthermore, an extension to our work will also evaluate the use of transfer learning using a pre-trained network for the imbalanced dataset classification.

# References

[1] J. Y. Choi, J. M. Lee, and C. B. Sirlin, “CT and MR imaging diagnosis and staging of hepatocellular carcinoma: Part I. Development, growth, and spread: Key pathologic and imaging aspects,” *Radiology*, vol. 272, no. 3, pp. 635–654, 2014.

[2] and N. D. P. Kohn Christine G., Prianka Singh, Beata Korytowsky, Jonathan T. Caranfa, Jeffrey D. Miller, Bruce E. Sill, Alexander C. Marshall, “Humanistic and economic burden of hepatocellular carcinoma: systematic literature review,” *Am. J. Manag. Care*, vol. 25, no. 2, pp. 61–73, 2019.

[3] J. M. Llovet *et al.*, “EASL-EORTC Clinical Practice Guidelines: Management of hepatocellular carcinoma,” *J. Hepatol.*, vol. 56, no. 4, pp. 908–943, 2012.

[4] C. T. Mri and L. Core, “Core LI-RADS CT/MRI Diagnostic Table.”

[5] W. Schima and J. Heiken, “LI-RADS v2017 for liver nodules: How we read and report,” *Cancer Imaging*, vol. 18, no. 1, pp. 1–11, 2018.

[6] K. M. Elsayes *et al.*, “2017 version of LI-RADS for CT and MR imaging: An update,” *Radiographics*, vol. 37, no. 7, pp. 1994–2017, 2017.

[7] L. Meng, C. Wen, and G. Li, “Support vector machine based liver cancer early detection using magnetic resonance images,” *2014 13th Int. Conf. Control Autom. Robot. Vision, ICARCV 2014*, vol. 2014, no. December, pp. 861–864, 2014.

[8] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “ImageNet classification with deep convolutional neural networks,” *Commun. ACM*, vol. 60, no. 6, pp. 84–90, 2017.

[9] A. Hosny, C. Parmar, J. Quackenbush, L. H. Schwartz, H. J. W. L. Aerts, and H. H. Edu, “Artificial intelligence in radiology,” *Nat Rev Cancer*, vol. 18, no. 8, pp. 500–510, 2018.

[10] Y. Lecun, Y. Bengio, and G. Hinton, “Deep learning,” *Nature*, vol. 521, no. 7553, pp. 436–444, 2015.

[11] H. He and E. A. Garcia, “Learning from Imbalanced Data,” *IEEE Trans. Knowl. Data Eng.*, vol. 21, no. 9, pp. 1263–1284, 2009.

[12] N. Japkowicz, “The Class Imbalance Problem: Significance and Strategies,” *Proc. 2000 Int. Conf. Artif. Intell.*, pp. 111--117, 2000.

[13] J. A. Marrero *et al.*, “Diagnosis, Staging, and Management of Hepatocellular Carcinoma: 2018 Practice Guidance by the American Association for the Study of Liver Diseases,” *Hepatology*, vol. 68, no. 2, pp. 723–750, 2018.

[14] T. Hennedige and S. K. Venkatesh, “Imaging of hepatocellular carcinoma: Diagnosis, staging and treatment monitoring,” *Cancer Imaging*, vol. 12, no. 3, pp. 530–547, 2012.

[15] S. Mittal and H. B. El-Serag, “Epidemiology of HCC: Consider the Population,” *J. Clin. Gastroenterol.*, vol. 47, no. 0, pp. 1–10, 2013.

[16] G. Baffy, E. M. Brunt, and S. H. Caldwell, “Hepatocellular carcinoma in non-alcoholic fatty liver disease: An emerging menace,” *J. Hepatol.*, vol. 56, no. 6, pp. 1384–1391, 2012.

[17] S. J. and T. Price, “基因的改变NIH Public Access,” *Bone*, vol. 23, no. 1, pp. 1–7, 2008.

[18] T. V. Bartolotta, A. Taibbi, M. Midiri, and R. Lagalla, “Contrast-enhanced ultrasound of hepatocellular carcinoma: Where do we stand?,” *Ultrasonography*, vol. 38, no. 3, pp. 200–214, 2019.

[19] S. Rossi *et al.*, “Contrast-enhanced ultrasonography and spiral computed tomography in the detection and characterization of portal vein thrombosis complicating hepatocellular carcinoma,” *Eur. Radiol.*, vol. 18, no. 8, pp. 1749–1756, 2008.

[20] H. J. Jang, K. K. Tae, P. N. Burns, and S. R. Wilson, “Enhancement patterns of hepatocellular carcinoma at contrast-enhanced US: Comparison with histologic differentiation,” *Radiology*, vol. 244, no. 3, pp. 898–906, 2007.

[21] T. V. Bartolotta, A. Taibbi, D. Picone, A. Anastasi, M. Midiri, and R. Lagalla, “Detection of liver metastases in cancer patients with geographic fatty infiltration of the liver: The added value of contrast-enhanced sonography,” *Ultrasonography*,

vol. 36, no. 2, pp. 160–169, 2017.

[22] S. R. Digumarthy, D. V. Sahani, and S. Saini, “MRI in detection of hepatocellular carcinoma (HCC),” *Cancer Imaging*, vol. 5, no. 1, pp. 20–24, 2005.

[23] M. S. Park *et al.*, “Hepatocellular carcinoma: Detection with diffusion-weighted versus contrast-enhanced magnetic resonance imaging in pretransplant patients,” *Hepatology*, vol. 56, no. 1, pp. 140–148, 2012.

[24] N. Albiin and N. C. Mri, “MRI of Focal Liver Lesions,” vol. 1, no. 909, pp. 107–116, 2012.

[25] M. Bruegel and E. J. Rummeny, “Hepatic metastases: Use of diffusion-weighted echo-planar imaging,” *Abdom. Imaging*, vol. 35, no. 4, pp. 454–461, 2010.

[26] A. D. Hardie *et al.*, “Diagnosis of liver metastases: Value of diffusion-weighted MRI compared with gadolinium-enhanced MRI,” *Eur. Radiol.*, vol. 20, no. 6, pp. 1431–1441, 2010.

[27] J. S. Yu, Y. H. Kim, and N. M. Rofsky, “Dynamic subtraction magnetic resonance imaging of cirrhotic liver: Assessment of high signal intensity lesions on nonenhanced T1-weighted images,” *J. Comput. Assist. Tomogr.*, vol. 29, no. 1, pp. 51–58, 2005.

[28] T. Haruyama *et al.*, “Gadolinium-based Contrast Agent Accumulates in the Brain Even in Subjects without Severe Renal Dysfunction: Evaluation of Autopsy Brain Specimens with Inductively Coupled Plasma Mass Spectroscopy,” *Radiology*, vol.

276, no. 1, 2015.

[29] K. Lertpipopmetha, T. Tubtawee, T. Piratvisuth, and N. Chamroonkul, "Comparison between computer tomography and magnetic resonance imaging in the diagnosis of small hepatocellular carcinoma," *Asian Pacific J. Cancer Prev.*, vol. 17, no. 11, pp. 4805–4811, 2016.

[30] A. Forner *et al.*, "Diagnosis of hepatic nodules 20 mm or smaller in cirrhosis: Prospective validation of the noninvasive diagnostic criteria for hepatocellular carcinoma," *Hepatology*, vol. 47, no. 1, pp. 97–104, 2008.

[31] Z. Sparchez and T. Mocan, "Contemporary role of liver biopsy in hepatocellular carcinoma," *World J. Hepatol.*, vol. 10, no. 7, pp. 452–461, 2018.

[32] L. Di Tommaso *et al.*, "Role of liver biopsy in hepatocellular carcinoma," *World J. Gastroenterol.*, vol. 25, no. 40, pp. 6041–6052, 2019.

[33] V. Chernyak *et al.*, "Liver Imaging Reporting and Data System (LI-RADS) version 2018: Imaging of hepatocellular carcinoma in at-risk patients," *Radiology*, vol. 289, no. 3, pp. 816–830, 2018.

[34] D. Vernuccio, F., Cannella, R., Meyer, M., Choudhoury, K.R., Gonzáles, F., Schwartz, F.R., Gupta, R.T., Bashir, M.R., Furlan, A. and Marin, "LI-RADS: Diagnostic Performance of Hepatobiliary Phase Hypointensity and Major Imaging Features of LR-3 and LR-4 Lesions Measuring 10–19 mm With Arterial Phase," *Am. J. Roentgenol.*, vol. 213, no. August, pp. 57–65, 2019.

[35] K. Suzuki, “Overview of deep learning in medical imaging,” *Radiol. Phys. Technol.*, vol. 10, no. 3, pp. 257–273, 2017.

[36] K. G. A. Gilhuijs, M. L. Giger, and U. Bick, “Computerized analysis of breast lesions in three dimensions using dynamic magnetic-resonance imaging,” *Med. Phys.*, vol. 25, no. 9, pp. 1647–1654, 1998.

[37] A. Masood *et al.*, “Computer-Assisted Decision Support System in Pulmonary Cancer detection and stage classification on CT images,” *Journal of Biomedical Informatics*, vol. 79. pp. 117–128, 2018.

[38] J. Levman, T. Leung, P. Causer, and A. L. Martel, “Author Manuscript / Manuscrit d ’ auteur breast lesions by support vector machines,” *Image (Rochester, N.Y.)*, vol. 27, no. 5, pp. 688–696, 2010.

[39] A. V. Faria, K. Oishi, S. Yoshida, A. Hillis, M. I. Miller, and S. Mori, “Content-based image retrieval for brain MRI: An image-searching engine and population-based analysis to utilize past clinical data for future diagnosis,” *NeuroImage Clin.*, vol. 7, pp. 367–376, 2015.

[40] and C. D. Evangelia I. Zacharakia Sumei Wanga, Sanjeev Chawlaa, Dong Soo Yooa, Ronald Wolfa, Elias R. Melhema, “Classification of brain tumor type and grade using MRI texture and shape in a machine learning scheme,” *Magn Reson Med*, vol. 62, no. 6, pp. 1609–1618, 2009.

[41] M. F. Siddiqui, G. Mujtaba, A. W. Reza, and L. Shuib, “Multi-class disease

classification in brain MRIs using a computer-aided diagnostic system," *Symmetry (Basel).*, vol. 9, no. 3, pp. 1–14, 2017.

[42] S. Chaplot, L. M. Patnaik, and N. R. Jagannathan, "Classification of magnetic resonance brain images using wavelets as input to support vector machine and neural network," *Biomed. Signal Process. Control*, vol. 1, no. 1, pp. 86–92, 2006.

[43] M. N. I. Qureshi, B. Min, H. J. Jo, and B. Lee, "Multiclass classification for the differential diagnosis on the ADHD subtypes using recursive feature elimination and hierarchical extreme learning machine: Structural MRI study," *PLoS One*, vol. 11, no. 8, pp. 1–20, 2016.

[44] X. Lin *et al.*, "A support vector machine-recursive feature elimination feature selection method based on artificial contrast variables and mutual information," *J. Chromatogr. B Anal. Technol. Biomed. Life Sci.*, vol. 910, pp. 149–155, 2012.

[45] N. Zhang, S. Ruan, S. Lebonvallet, Q. Liao, and Y. Zhu, "Multi-kernel SVM based classification for brain tumor segmentation of MRI multi-sequence," *Proc. - Int. Conf. Image Process. ICIP*, no. May 2014, pp. 3373–3376, 2009.

[46] W. Chen, M. L. Giger, L. Lan, and U. Bick, "Computerized interpretation of breast MRI: Investigation of enhancement-variance dynamics," *Med. Phys.*, vol. 31, no. 5, pp. 1076–1082, 2004.

[47] C. Sun *et al.*, "Automatic segmentation of liver tumors from multiphase contrast-enhanced CT images based on FCNs," *Artif. Intell. Med.*, vol. 83, pp. 58–66, 2017.

[48] G. Z. Li, J. Yang, C. Z. Ye, and D. Y. Geng, “Degree prediction of malignancy in brain glioma using support vector machines,” *Comput. Biol. Med.*, vol. 36, no. 3, pp. 313–325, 2006.

[49] J. Jiang, P. Trundle, and J. Ren, “Medical image analysis with artificial neural networks,” *Comput. Med. Imaging Graph.*, vol. 34, no. 8, pp. 617–631, 2010.

[50] Z. Akkus, A. Galimzianova, A. Hoogi, D. L. Rubin, and B. J. Erickson, “Deep Learning for Brain MRI Segmentation: State of the Art and Future Directions,” *J. Digit. Imaging*, vol. 30, no. 4, pp. 449–459, 2017.

[51] K. Yasaka, H. Akai, O. Abe, and S. Kiryu, “Deep learning with CNN showed high diagnostic performance in differentiation of liver masses at dynamic CT,” *Radiology*, vol. 286, no. 3—March, pp. 887–896, 2018.

[52] E. I. Zacharaki, V. G. Kanas, and C. Davatzikos, “Investigating machine learning techniques for MRI-based classification of brain neoplasms,” *Int. J. Comput. Assist. Radiol. Surg.*, vol. 6, no. 6, pp. 821–828, 2011.

[53] K. Tzartzeva, J. Obi, J. A. M. , Nicole E. Rich1, Neehar D. Parikh2, and 4 Adam Yopp3, Akbar Waljee2, and Amit G. Singal1, “Surveillance Imaging and Alpha Fetoprotein for Early Detection of Hepatocellular Carcinoma in Patients With Cirrhosis: A Meta-analysis,” *Physiol. Behav.*, vol. 176, no. 3, pp. 139–148, 2019.

[54] M. Nawaz, A. A., and T. Hassan, “Multi-Class Breast Cancer Classification using Deep Learning Convolutional Neural Network,” *Int. J. Adv. Comput. Sci. Appl.*, vol. 9, no.

6, pp. 316–322, 2018.

[55] P. Lakhani and B. Sundaram, “Deep learning at chest radiography: Automated classification of pulmonary tuberculosis by using convolutional neural networks,” *Radiology*, vol. 284, no. 2, pp. 574–582, 2017.

[56] H. Mohsen, E.-S. A. El-Dahshan, E.-S. M. El-Horbaty, and A.-B. M. Salem, “Classification using deep learning neural networks for brain tumors,” *Futur. Comput. Informatics J.*, vol. 3, no. 1, pp. 68–71, 2017.

[57] T. Noguchi *et al.*, “Artificial intelligence using neural network architecture for radiology (AINNAR): classification of MR imaging sequences,” *Jpn. J. Radiol.*, vol. 36, no. 12, pp. 691–697, 2018.

[58] K. R. Cave and N. P. Bichot, “Visuospatial attention: Beyond a spotlight model,” *Psychon. Bull. Rev.*, vol. 6, no. 2, pp. 204–223, 1999.

[59] Y. LeCun, P. Haffner, L. Bottou, and Y. Bengio, “Object recognition with gradient-based learning,” *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 1681, no. 0, pp. 319–345, 1999.

[60] W. Wei, J. Li, L. Cao, Y. Ou, and J. Chen, “Effective detection of sophisticated online banking fraud on extremely imbalanced data,” *World Wide Web*, vol. 16, no. 4, pp. 449–475, 2013.

[61] M. Herland, T. M. Khoshgoftaar, and R. A. Bauder, “Big Data fraud detection using multiple medicare data sources,” *J. Big Data*, vol. 5, no. 1, pp. 1–21, 2018.

[62] R. A. Bauder and T. M. Khoshgoftaar, “The effects of varying class distribution on learner behavior for medicare fraud detection with imbalanced big data,” *Heal. Inf. Sci. Syst.*, vol. 6, no. 1, pp. 1–14, 2018.

[63] J. Davis and M. Goadrich, “The relationship between Precision-Recall and ROC curves,” *Proc. 23rd Int. Conf. Mach. Learn. - ICML*, vol. 6, 2006.

[64] J. Van Hulse, T. M. Khoshgoftaar, and A. Napolitano, “Experimental perspectives on learning from imbalanced data,” *ACM Int. Conf. Proceeding Ser.*, vol. 227, no. January, pp. 935–942, 2007.

[65] W. P. K. N. V. C. K. W. B. Lawrence O. Hall, “SMOTE: Synthetic Minority Over-sampling Technique Nitesh,” *J. Artif. Intell. Res.*, vol. 2009, no. Sept. 28, pp. 321–357, 2006.

[66] S. H. Khan, M. Hayat, M. Bennamoun, F. A. Sohel, and R. Togneri, “Cost-sensitive learning of deep feature representations from imbalanced data,” *IEEE Trans. Neural Networks Learn. Syst.*, vol. 29, no. 8, pp. 3573–3587, 2018.

[67] Q. Dong, S. Gong, and X. Zhu, “Imbalanced Deep Learning by Minority Class Incremental Rectification,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 41, no. 6, pp. 1367–1381, 2019.

[68] M. Kubat and S. Matwin, “Addressing the Curse of Imbalanced Training Sets: One Sided Selection,” *Icml*, vol. 97, pp. 179–186, 1997.

[69] J. Stefanowski and S. Wilk, “Selective pre-processing of imbalanced data for

improving classification performance," *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 5182 LNCS, pp. 283–292, 2008.

[70] J. Laurikkala, "Improving Identification of Difficult Small Classes By," *Inf. Sci. (Ny).*, 2001.

[71] P. Hensman and D. Masko, "The Impact of Imbalanced Training Data for Convolutional Neural Networks," *PhD*, 2015.

[72] M. Buda, A. Maki, and M. A. Mazurowski, "A systematic study of the class imbalance problem in convolutional neural networks," *Neural Networks*, vol. 106, pp. 249–259, 2018.

[73] J. Bacardit, E. Bernadó-Mansilla, and M. V. Butz, "Learning classifier systems: Looking back and glimpsing ahead," *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 4998 LNAI, pp. 1–21, 2008.

[74] A. Orriols-Puig and E. Bernadó-Mansilla, "Evolutionary rule-based systems for imbalanced data sets," *Soft Comput.*, vol. 13, no. 3, pp. 213–225, 2009.

[75] E. C. Orenstein, O. Beijbom, E. E. Peacock, and H. M. Sosik, "WHOI-Plankton- A Large Scale Fine Grained Visual Recognition Benchmark Dataset for Plankton Classification," 2015.

[76] H. Lee, M. Park, and J. Kim, "Plankton classification on imbalanced large scale database via convolutional neural networks with transfer learning," *Proc. - Int.*

*Conf. Image Process. ICIP*, vol. 2016-Augus, pp. 3713–3717, 2016.

[77] S. Pouyanfar *et al.*, “Dynamic Sampling in Convolutional Neural Networks for Imbalanced Data Classification,” *Proc. - IEEE 1st Conf. Multimed. Inf. Process. Retrieval, MIPR 2018*, no. June, pp. 112–117, 2018.

[78] S. Wang, W. Liu, J. Wu, L. Cao, Q. Meng, and P. J. Kennedy, “Training deep neural networks on imbalanced data sets,” *Proc. Int. Jt. Conf. Neural Networks*, vol. 2016-Octob, pp. 4368–4374, 2016.

[79] T. Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollar, “Focal Loss for Dense Object Detection,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 42, no. 2, pp. 318–327, 2020.

[80] R. Anand, K. G. Mehrotra, C. K. Mohan, and S. Ranka, “An Improved Algorithm for Neural Network Classification of Imbalanced Training Sets,” *IEEE Trans. Neural Networks*, vol. 4, no. 6, pp. 962–969, 1993.

[81] R. Girshick, J. Donahue, T. Darrell, J. Malik, U. C. Berkeley, and J. Malik, “Rich feature hierarchies for accurate object detection and semantic segmentation,” *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, vol. 1, p. 5000, 2014.

[82] P. Felzenszwalb, R. Girshick, D. McAllester, and D. Ramanan, “Visual object detection with deformable part models,” *Commun. ACM*, vol. 56, no. 9, pp. 97–105, 2013.

[83] H. S. Nemoto K, Hamaguchi R, Imaizumi T, “Classification of Rare Building Change

Using CNN with Multi-Class Focal Loss," *IGARSS 2018 - 2018 IEEE Int. Geosci. Remote Sens. Symp. Val.*, pp. 4663-4666, 2018.

[84] J. Cid-Sueiro, J. I. Arribas, S. Urbán-Muñoz, and A. R. Figueiras-Vidal, "Cost functions to estimate a posteriori probabilities in multiclass problems," *IEEE Trans. Neural Networks*, vol. 10, no. 3, pp. 645–656, 1999.

[85] L. Engstrom, B. Tran, D. Tsipras, L. Schmidt, and A. Madry, "Exploring the landscape of spatial robustness," *36th Int. Conf. Mach. Learn. ICML 2019*, vol. 2019-June, pp. 3218–3238, 2019.

[86] L. Taylor and G. Nitschke, "Improving Deep Learning using Generic Data Augmentation," 2017.

[87] K. Chatfield, K. Simonyan, A. Vedaldi, and A. Zisserman, "Return of the devil in the details: Delving deep into convolutional nets," *BMVC 2014 - Proc. Br. Mach. Vis. Conf. 2014*, pp. 1–11, 2014.

[88] S. Dieleman, K. W. Willett, and J. Dambre, "Rotation-invariant convolutional neural networks for galaxy morphology prediction," *Mon. Not. R. Astron. Soc.*, vol. 450, no. 2, pp. 1441–1459, 2015.

[89] Y. D. Zhang *et al.*, "Image based fruit category classification by 13-layer deep convolutional neural network and data augmentation," *Multimed. Tools Appl.*, vol. 78, no. 3, pp. 3613–3632, 2019.

[90] J. Ding, B. Chen, H. Liu, and M. Huang, "Convolutional Neural Network with Data

Augmentation for SAR Target Recognition," *IEEE Geosci. Remote Sens. Lett.*, vol. 13, no. 3, pp. 364–368, 2016.

[91] S. Raju, M. S. Nasir, and T. M. Devi, "Filtering Techniques to reduce Speckle Noise and Image Quality Enhancement methods on Satellite Images," *IOSR J. Comput. Eng.*, vol. 15, no. 4, pp. 10–15, 2013.

[92] Z. Hussain, F. Gimenez, D. Yi, and D. Rubin, "Differential Data Augmentation Techniques for Medical Imaging Classification Tasks," *AMIA ... Annu. Symp. proceedings. AMIA Symp.*, vol. 2017, pp. 979–984, 2017.

[93] L. Perez and J. Wang, "The Effectiveness of Data Augmentation in Image Classification using Deep Learning," 2017.

[94] J. Y. Zhu, T. Park, P. Isola, and A. A. Efros, "Unpaired Image-to-Image Translation Using Cycle-Consistent Adversarial Networks," *Proc. IEEE Int. Conf. Comput. Vis.*, vol. 2017-Octob, pp. 2242–2251, 2017.

[95] E. D. Cubuk, B. Zoph, D. Mane, V. Vasudevan, and Q. V. Le, "Autoaugment: Learning augmentation strategies from data," *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, vol. 2019-June, no. Section 3, pp. 113–123, 2019.

[96] A. J. Ratner, H. R. Ehrenberg, Z. Hussain, J. Dunnmon, and C. Ré, "Learning to compose domain-specific transformations for data augmentation," *Adv. Neural Inf. Process. Syst.*, vol. 2017-Decem, no. Nips, pp. 3237–3247, 2017.

[97] Y. Netzer, T. Wang, A. Coates, A. Bissacco, B. Wu, and A. Y. Ng, "Reading Digits in

Natural Images with Unsupervised Feature Learning," *Vopr. Neirokhir.*, vol. 16, no. 5, pp. 9–13, 1952.

[98] S. Lim, I. Kim, T. Kim, C. Kim, and S. Kim, "Fast AutoAugment," no. NeurIPS, 2019.

[99] H. C. Shin *et al.*, "Medical image synthesis for data augmentation and anonymization using generative adversarial networks," *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 11037 LNCS, pp. 1–11, 2018.

[100] P. Isola, J. Y. Zhu, T. Zhou, and A. A. Efros, "Image-to-image translation with conditional adversarial networks," *Proc. - 30th IEEE Conf. Comput. Vis. Pattern Recognition, CVPR 2017*, vol. 2017-Janua, pp. 5967–5976, 2017.

[101] W. Ding, D. Y. Huang, Z. Chen, X. Yu, and W. Lin, "Facial action recognition using very deep networks for highly imbalanced class distribution," *Proc. - 9th Asia-Pacific Signal Inf. Process. Assoc. Annu. Summit Conf. APSIPA ASC 2017*, vol. 2018-Febru, no. December, pp. 1368–1372, 2018.

[102] A. Khan, A. Sohail, U. Zahoora, and A. S. Qureshi, "A survey of the recent architectures of deep convolutional neural networks," *Artif. Intell. Rev.*, pp. 1–70, 2020.

[103] Y. LeCun, C. Cortes, L. Bottou, and L. Jackel, "Comparison of Learning Algorithms for Handwriting Digit Recognition," *Int. Conf. Artif. Neural Networks*, pp. 53–60, 1995.

[104] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “ImageNet Classification with Deep Convolutional Neural Networks,” *Adv. Neural Inf. Process. Syst.*, pp. 1–9, 2012.

[105] R. Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I. and Salakhutdinov, “Dropout: A Simple Way to Prevent Neural Networks from Overfitting Nitish,” *J. Mach. Learn. Res.*, vol. 15, no. 1, pp. 1929–1958, 2014.

[106] K. Simonyan and A. Zisserman, “Very deep convolutional networks for large-scale image recognition,” *3rd Int. Conf. Learn. Represent. ICLR 2015 - Conf. Track Proc.*, pp. 1–14, 2015.

[107] X. Glorot and Y. Bengio, “Understanding the difficulty of training deep feedforward neural networks,” *J. Mach. Learn. Res.*, vol. 9, pp. 249–256, 2010.

[108] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, “Rethinking the Inception Architecture for Computer Vision,” *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, vol. 2016-Decem, pp. 2818–2826, 2016.

[109] M. Lin, Q. Chen, and S. Yan, “Network in network,” *2nd Int. Conf. Learn. Represent. ICLR 2014 - Conf. Track Proc.*, pp. 1–10, 2014.

[110] A. R. Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, “Going Deeper with Convolutions,” *IEEE Conf. Comput. Vis. Pattern Recognit.*, pp. 1–9, 2015.

[111] K. Grm, V. Struc, A. Artiges, M. Caron, and H. K. Ekenel, “Strengths and weaknesses of deep learning models for face recognition against image degradations,” *IET*

*Biometrics*, vol. 7, no. 1, pp. 81–89, 2018.

[112] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, vol. 2016-Decem, pp. 770–778, 2016.

[113] R. K. Srivastava, K. Greff, and J. Schmidhuber, "Highway Networks," 2015.

[114] K. He and J. Sun, "Convolutional neural networks at constrained time cost," *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, vol. 07-12-June, pp. 5353–5360, 2015.

[115] S. H. Hasanpour, M. Rouhani, M. Fayyaz, and M. Sabokrou, "Lets keep it simple, Using simple architectures to outperform deeper and more complex architectures," pp. 1–18, 2016.

[116] Q. Li, W. Cai, X. Wang, Y. Zhou, D. D. Feng, and M. Chen, "Medical image classification with convolutional neural network," *2014 13th Int. Conf. Control Autom. Robot. Vision, ICARCV 2014*, vol. 2014, no. December, pp. 844–848, 2014.

[117] A. Depeursinge, A. Vargas, A. Platon, A. Geissbuhler, P. A. Poletti, and H. Müller, "Building a reference multimedia database for interstitial lung diseases," *Comput. Med. Imaging Graph.*, vol. 36, no. 3, pp. 227–238, 2012.

[118] E. Montagnon *et al.*, "Deep learning workflow in radiology: a primer," *Insights Imaging*, vol. 11, no. 1, 2020.

[119] S. Ozturk, U. Ozkaya, B. Akdemir, and L. Seyfi, "Convolution kernel size effect on

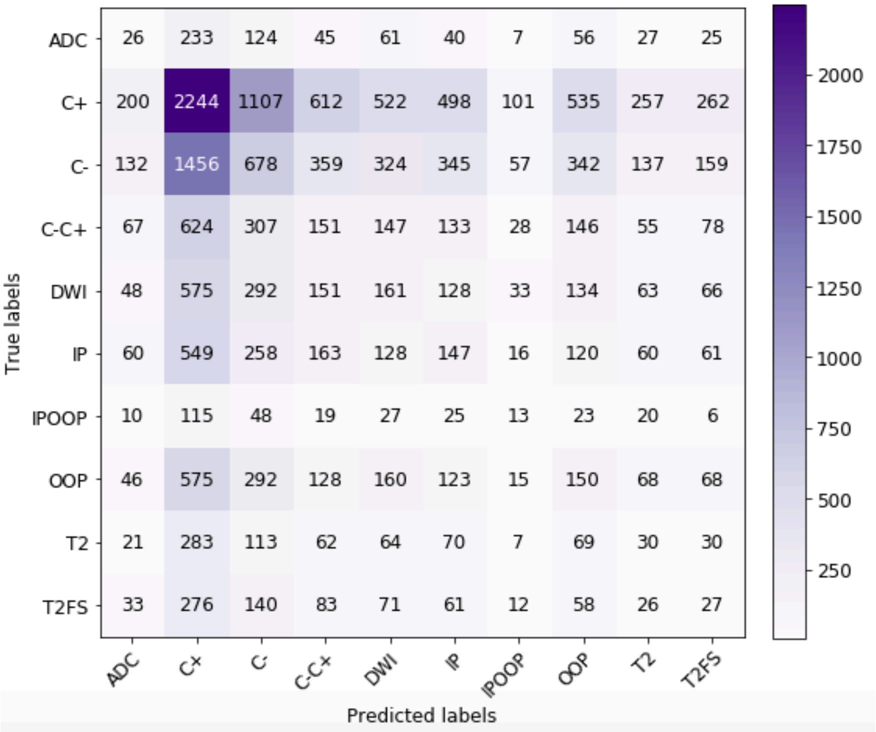
convolutional neural network in histopathological image processing applications," *2018 Int. Symp. Fundam. Electr. Eng. ISFEE 2018*, 2018.

[120] R. Yamashita, M. Nishio, R. K. G. Do, and K. Togashi, "Convolutional neural networks: an overview and application in radiology," *Insights Imaging*, vol. 9, no. 4, pp. 611–629, 2018.

# Appendix

## A.1 Classification Predictions of Model A, B and C on Shuffled Test Set

Here, we report classification predictions when the test set is shuffled prior to making predictions for classes.



| True labels \ Predicted labels | ADC | C+ | C- | C-C+ | DWI | IP | IPOOP | OOP | T2 | T2FS |
|---|---|---|---|---|---|---|---|---|---|---|
| ADC | 26 | 233 | 124 | 45 | 61 | 40 | 7 | 56 | 27 | 25 |
| C+ | 200 | 2244 | 1107 | 612 | 522 | 498 | 101 | 535 | 257 | 262 |
| C- | 132 | 1456 | 678 | 359 | 324 | 345 | 57 | 342 | 137 | 159 |
| C-C+ | 67 | 624 | 307 | 151 | 147 | 133 | 28 | 146 | 55 | 78 |
| DWI | 48 | 575 | 292 | 151 | 161 | 128 | 33 | 134 | 63 | 66 |
| IP | 60 | 549 | 258 | 163 | 128 | 147 | 16 | 120 | 60 | 61 |
| IPOOP | 10 | 115 | 48 | 19 | 27 | 25 | 13 | 23 | 20 | 6 |
| OOP | 46 | 575 | 292 | 128 | 160 | 123 | 15 | 150 | 68 | 68 |
| T2 | 21 | 283 | 113 | 62 | 64 | 70 | 7 | 69 | 30 | 30 |
| T2FS | 33 | 276 | 140 | 83 | 71 | 61 | 12 | 58 | 26 | 27 |

Figure A 1: Model A (sequence dataset) confusion matrix results

Table A 1: Classification report for the above Model A (sequence dataset)

| | precision | recall | F1-score | support |
|---|---|---|---|---|
| ADC | 0.04 | 0.04 | 0.04 | 644 |
| C+ | 0.32 | 0.35 | 0.34 | 6338 |
| C- | 0.2 | 0.17 | 0.18 | 3989 |
| C-C+ | 0.09 | 0.09 | 0.09 | 1736 |

| | | | | |
|---|---|---|---|---|
| DWI | 0.1 | 0.1 | 0.1 | 1651 |
| IP | 0.09 | 0.09 | 0.09 | 1562 |
| IPOOP | 0.04 | 0.04 | 0.04 | 306 |
| OOP | 0.09 | 0.09 | 0.09 | 1625 |
| T2 | 0.04 | 0.04 | 0.04 | 749 |
| T2FS | 0.03 | 0.03 | 0.03 | 787 |
| accuracy | | | 0.19 | 19387 |
| macro avg | 0.11 | 0.11 | 0.11 | 19387 |
| weighted avg | 0.18 | 0.19 | 0.18 | 19387 |

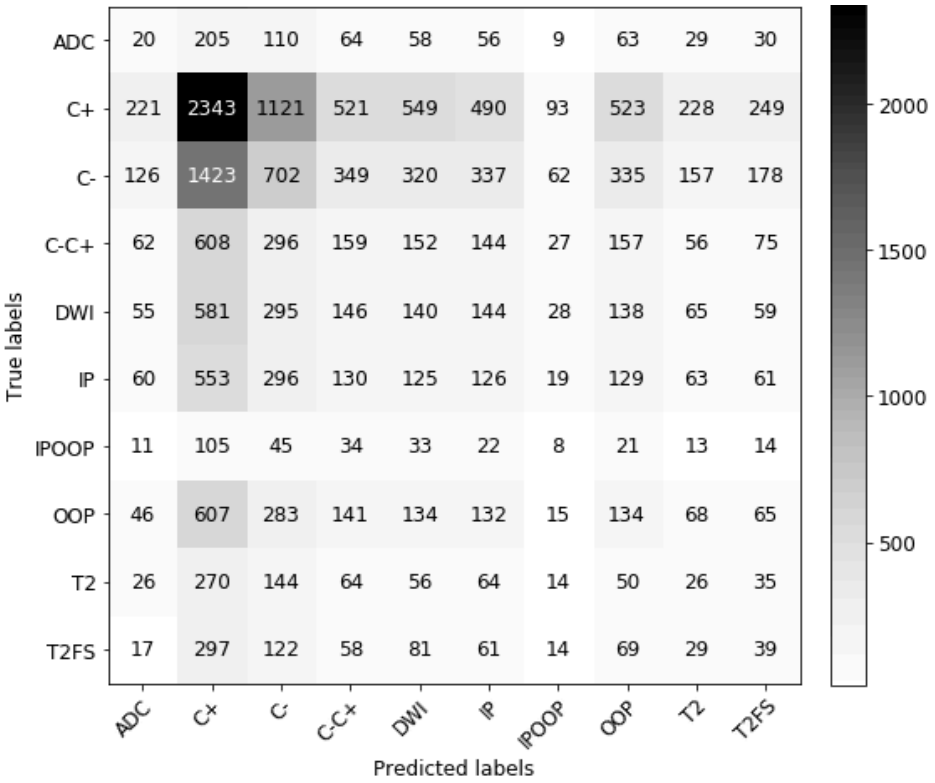


Figure A 2: Confusion matrix results for Model B (sequence dataset)

Table A 2: Classification report for Model B (sequence dataset)

| | precision | recall | F1-score | support |
|---|---|---|---|---|
| ADC | 0.03 | 0.03 | 0.03 | 644 |
| C+ | 0.32 | 0.37 | 0.34 | 6338 |
| C- | 0.21 | 0.16 | 0.18 | 3989 |
| C-C+ | 0.09 | 0.1 | 0.1 | 1736 |
| DWI | 0.08 | 0.08 | 0.08 | 1651 |

| | | | | |
|---|---|---|---|---|
| IP | 0.08 | 0.08 | 0.08 | 1562 |
| IPOOP | 0.01 | 0.01 | 0.01 | 306 |
| OOP | 0.08 | 0.09 | 0.09 | 1625 |
| T2 | 0.04 | 0.04 | 0.04 | 749 |
| T2FS | 0.05 | 0.05 | 0.05 | 787 |
| accuracy | | | 0.19 | 19387 |
| macro avg | 0.1 | 0.1 | 0.1 | 19387 |
| weighted avg | 0.18 | 0.19 | 0.18 | 19387 |

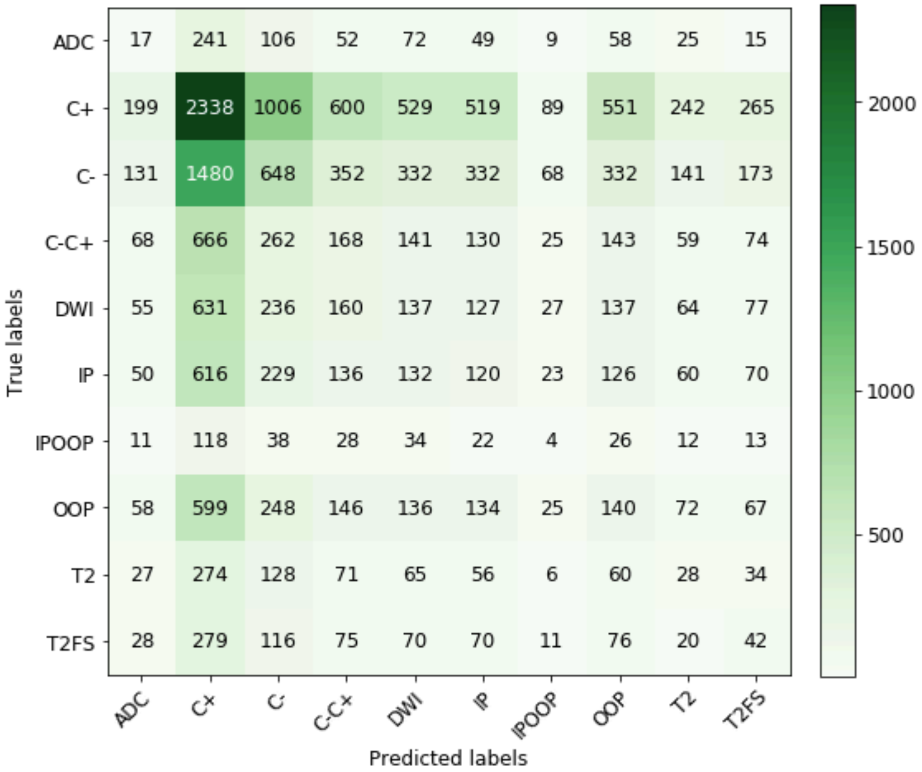


Figure A 3: Confusion matrix results for Model C (using augmented sequence dataset)

Table A 3: Classification report for Model C (augmented images)

| | precision | recall | F1-score | support |
|---|---|---|---|---|
| ADC | 0.03 | 0.03 | 0.03 | 644 |
| C+ | 0.34 | 0.37 | 0.35 | 6338 |
| C- | 0.21 | 0.18 | 0.19 | 3989 |
| C-C+ | 0.1 | 0.09 | 0.09 | 1736 |
| DWI | 0.08 | 0.08 | 0.08 | 1651 |

| | | | | |
|---|---|---|---|---|
| IP | 0.08 | 0.08 | 0.08 | 1562 |
| IPOOP | 0.03 | 0.03 | 0.03 | 306 |
| OOP | 0.08 | 0.08 | 0.08 | 1625 |
| T2 | 0.04 | 0.03 | 0.04 | 749 |
| T2FS | 0.05 | 0.05 | 0.05 | 787 |
| accuracy | | | 0.19 | 19387 |
| macro avg | 0.1 | 0.1 | 0.1 | 19387 |
| weighted avg | 0.19 | 0.19 | 0.19 | 19387 |

Table A 4: F1 score summary for all models. Test was shuffled prior to making predictions

| | F1-score | | |
|---|---|---|---|
| | Model A | Model B | Model C |
| ADC | 0.04 | 0.03 | 0.03 |
| C+ | 0.34 | 0.34 | 0.35 |
| C- | 0.18 | 0.18 | 0.19 |
| C-C+ | 0.09 | 0.1 | 0.09 |
| DWI | 0.1 | 0.08 | 0.08 |
| IP | 0.09 | 0.08 | 0.08 |
| IPOOP | 0.04 | 0.01 | 0.03 |
| OOP | 0.09 | 0.09 | 0.08 |
| T2 | 0.04 | 0.04 | 0.04 |
| T2FS | 0.03 | 0.05 | 0.05 |

| | | | |
|---|---|---|---|
| accuracy | 0.19 | 0.19 | 0.19 |
| macro avg | 0.11 | 0.1 | 0.1 |
| weighted avg | 0.18 | 0.18 | 0.19 |

From Figure A 1, Figure A 2 and Figure A 3it can be observed that the model fits to the training set distribution well but it cannot generalize on a shuffled test set when making predictions. Furthermore, tables A1, A2 and A3 show the F1-score reported for the T1-weighted post contrast images (C-, C+ and C-C+) is higher than the minority classes.

## A.2 Cross Validation Results

Five-fold cross validation was run using Model A on the imbalanced sequence dataset. This was run to experiment whether the imbalanced distribution in the training set would affect the model's ability to classify the minority classes.

From Figure A 4 and Figure A 5 it is observed that the model does not generalize well on the test set when the training set is imbalanced. The results show that the model suffers significantly when there are high number of samples for the majority class (T1-weighted post contrast C-, C+) as compared to the other sequences.

Furthermore, from Figure A 4 and Figure A 5 it is also observed that there is gradient saturation when the model weights get updated. This could also be referred to as the vanishing gradients problem which leads to no update to the network connections which look for particular features on the image.

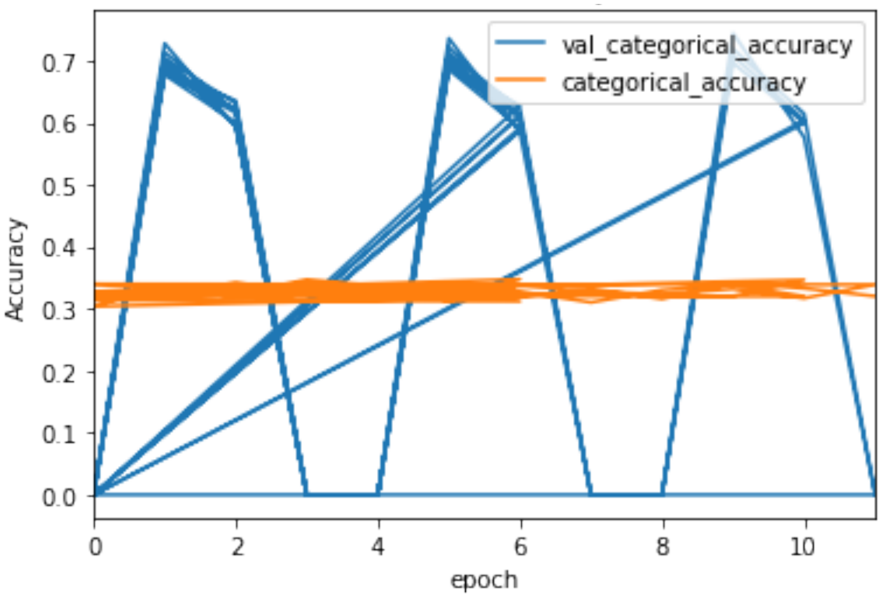


Figure A 4: Accuracy for all five folds during training Model A (baseline)

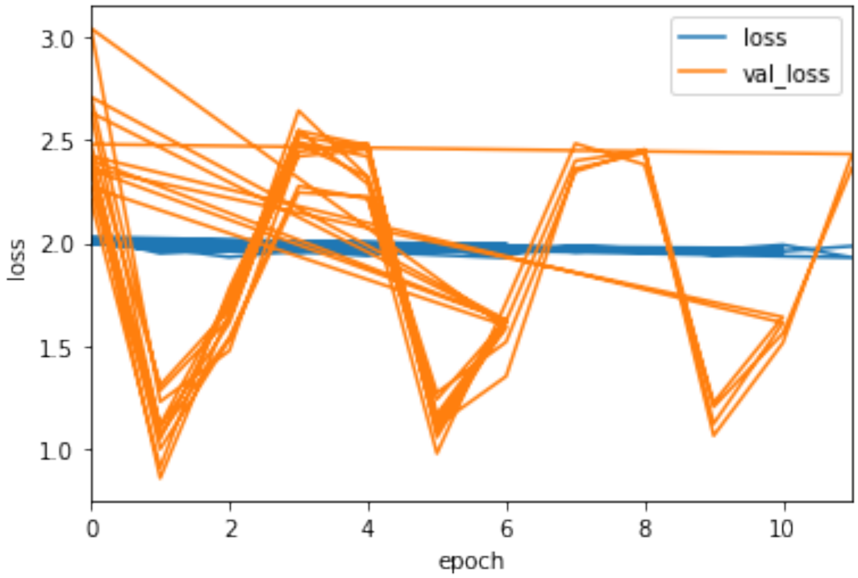


Figure A 5: Training loss plot as training epochs progressed for five-fold cross validation.

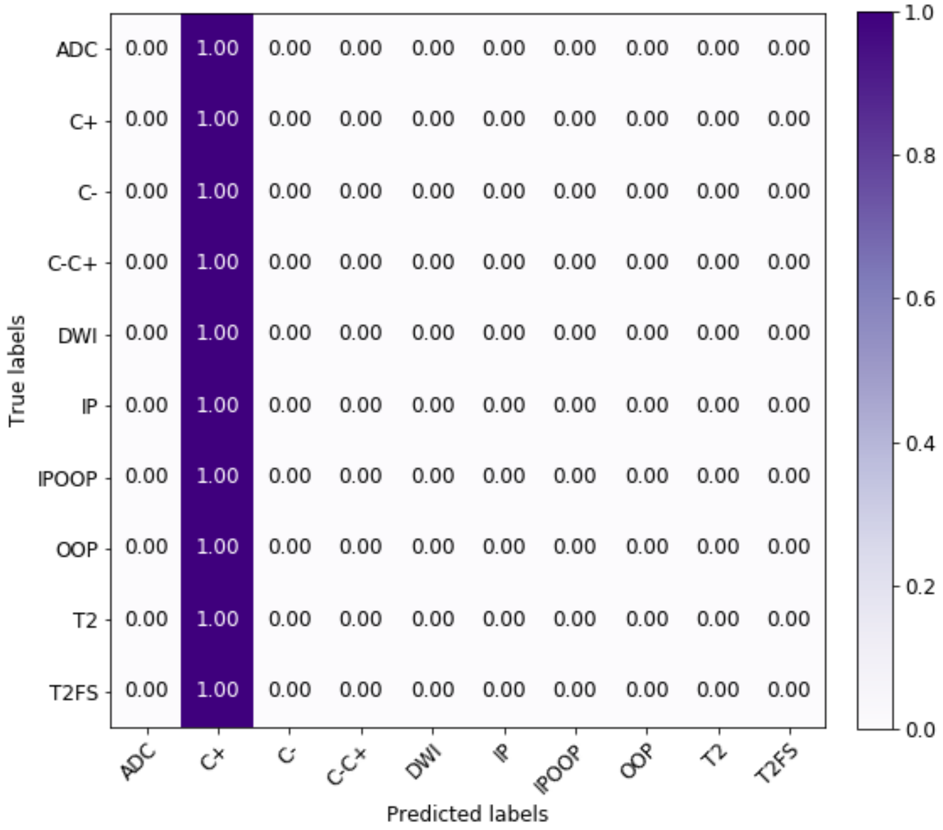


Figure A 6: Confusion matrix result for five-fold cross validation using Model A on the imbalanced sequence dataset.

Figure A 6 shows that the model is perfectly biased to the majority C+ class. For the confusion matrix result in Figure A 6 it should also be note that prediction probabilities for the other minority classes is not shown as the values beyond two decimal points are disregarded in the plot.

Cross validation performed on the imbalanced MRI sequence training set showed that the model performance of Model A suffers significantly when there is underrepresentation of other classes. This indicates that the model training should be performed on a balanced distribution among the classes in the training set. A limitation should be noted that the method did not use stratification of samples during each fold.

## A.3 MRI Phases Dataset Characteristics

In this section, dataset statistics for liver MRI phases from the T1-weighted post contrast sequences C-, C+ and C-C+ have been provided.

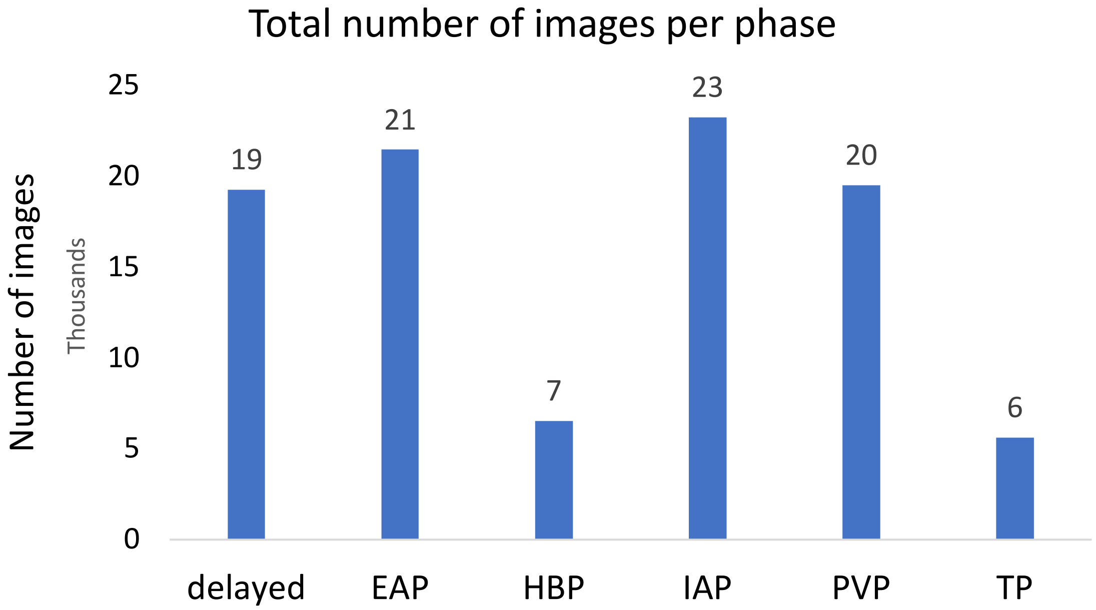


Figure A 7: Distribution of phases in post contrast sequences

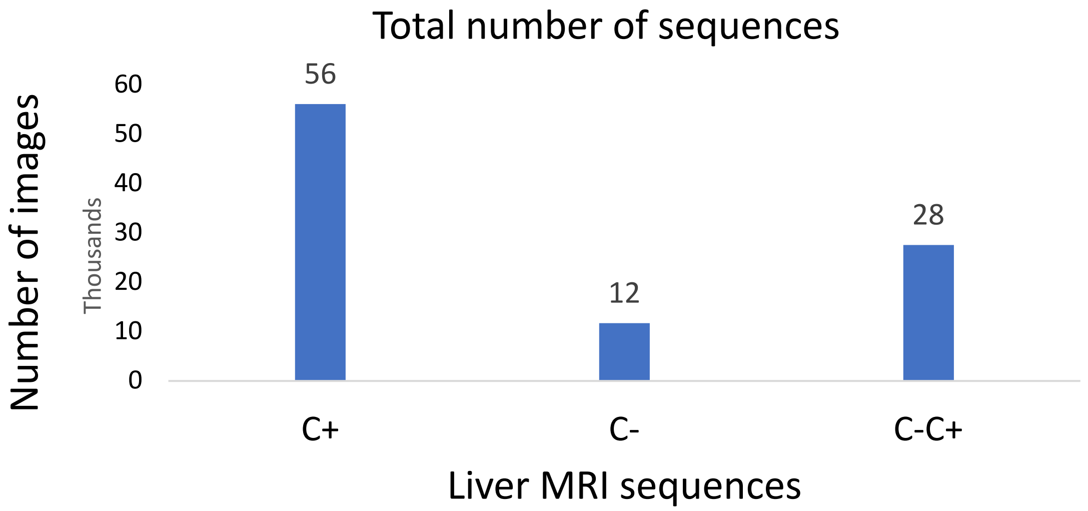


Figure A 8: Distribution of post contrast sequences in the phases dataset

In addition to the MRI sequence dataset, phase of enhancement is also considered for the LI-RADS application. As observed from Figure A 10, the phase of enhancement is determined from the administration of post contrast agent. These are captured in sequences shown in Figure A 10.

The phase of enhancement was obtained from the same post contrast sequences as was used in the sequence dataset (Figure 3: Distribution of images per MRI sequence. As compared to the T1-weighted post-contrast (C- and C+) phases, all other phases have significantly lower samples).

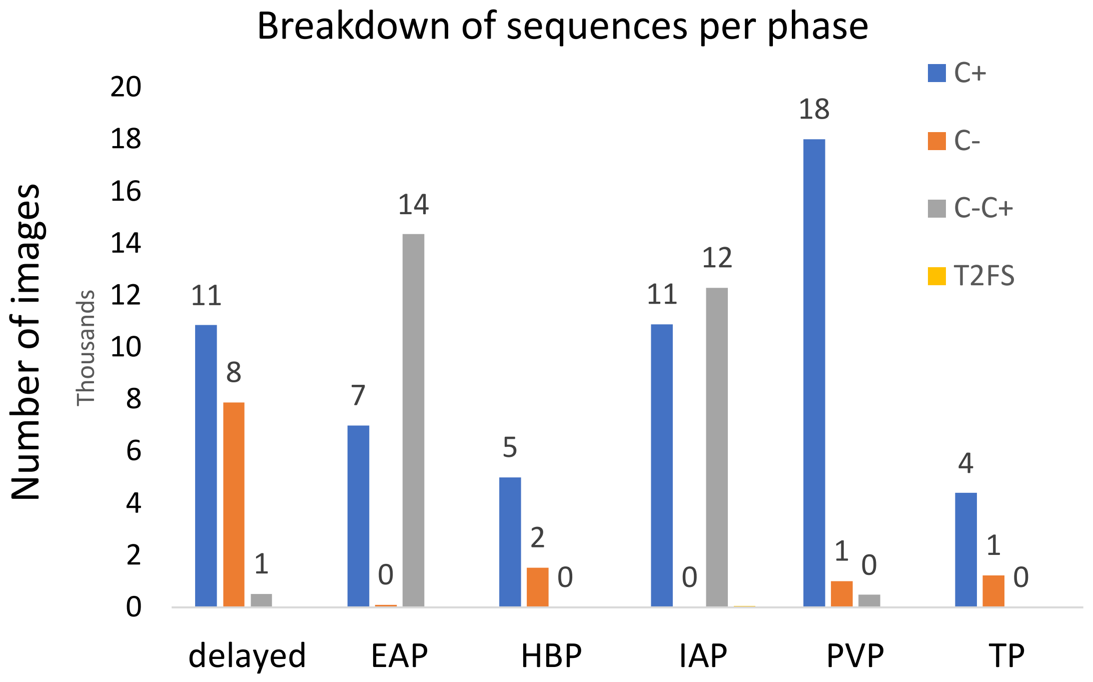


Figure A 9: Overall distribution of phases per post contrast sequence

As inferred from Figure A 9, C+ contains more images per phase of enhancement. T2FS only had 40 images for the delayed phase and not shown in the graph above.

In order to correctly diagnose a patient to a LI-RADS category post-contrast MRI phases (includes administration of contrast agent such as gadolinium) provide the most information

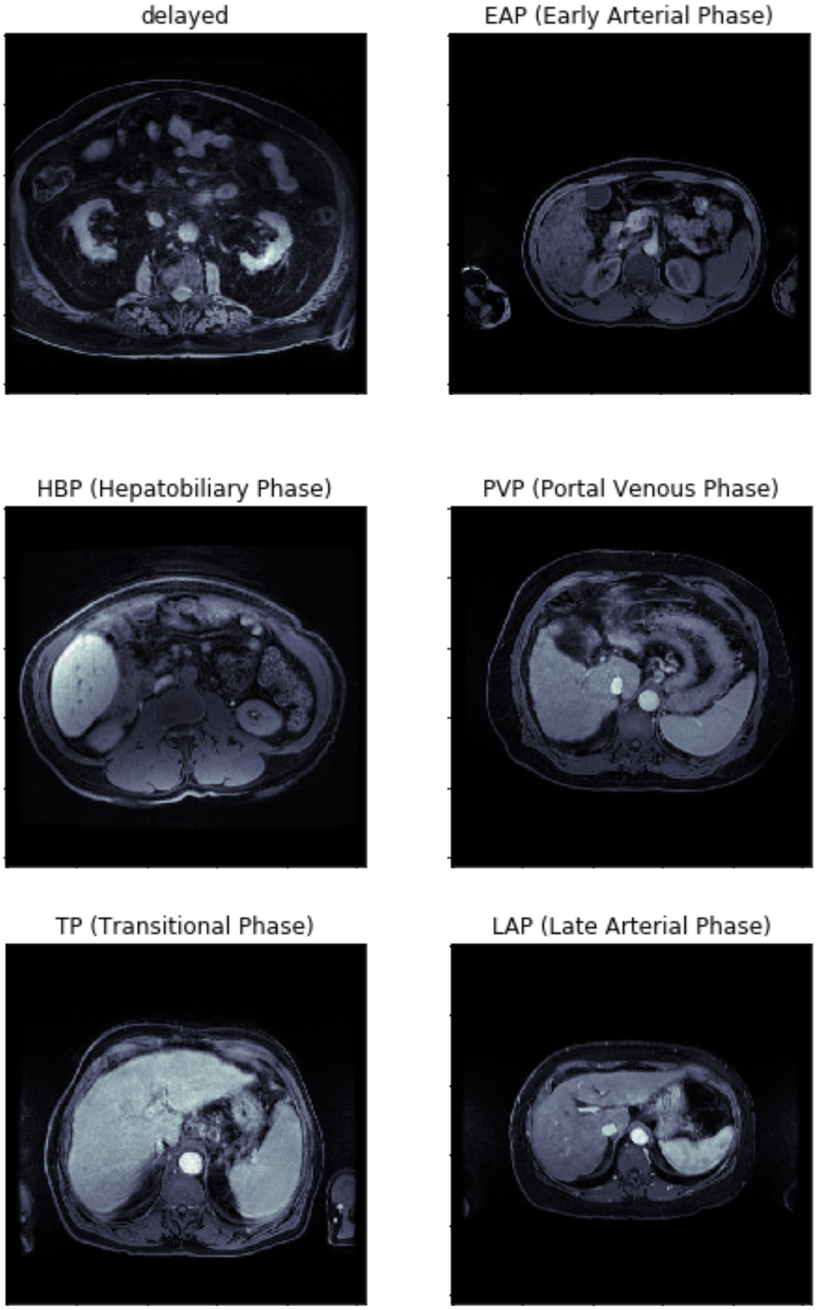

delayed
EAP (Early Arterial Phase)
HBP (Hepatobiliary Phase)
PVP (Portal Venous Phase)
TP (Transitional Phase)
LAP (Late Arterial Phase)

Figure A 10: Sample images of different phases

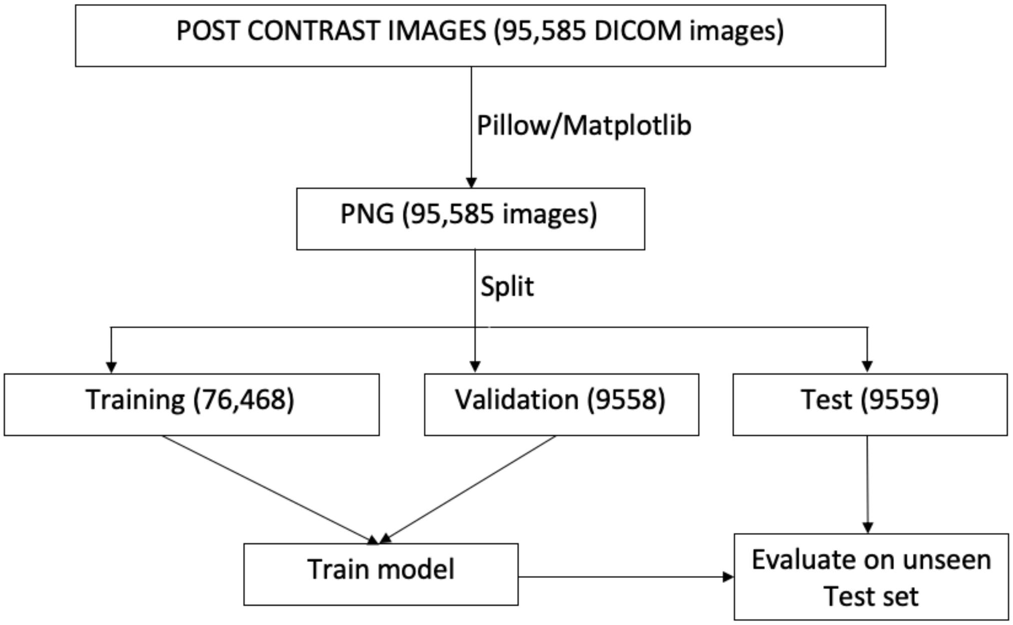


Figure A 11: Dataset preparation for phase of post contrast sequences (C+, C-, C-C+ and T2FS)

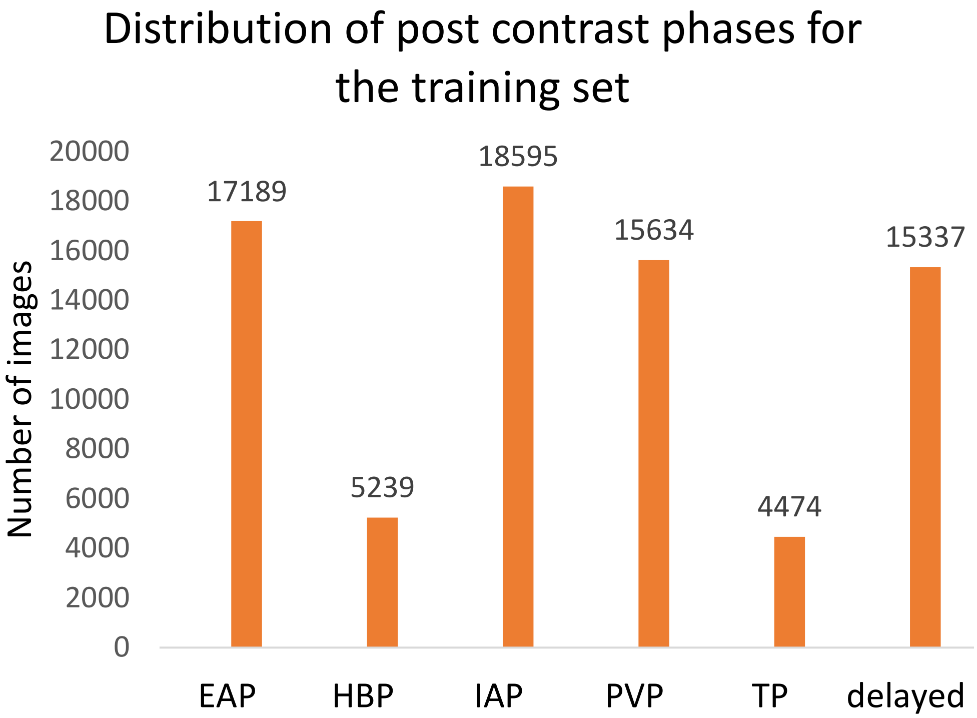


Figure A 12: Number of images per phase in the training set

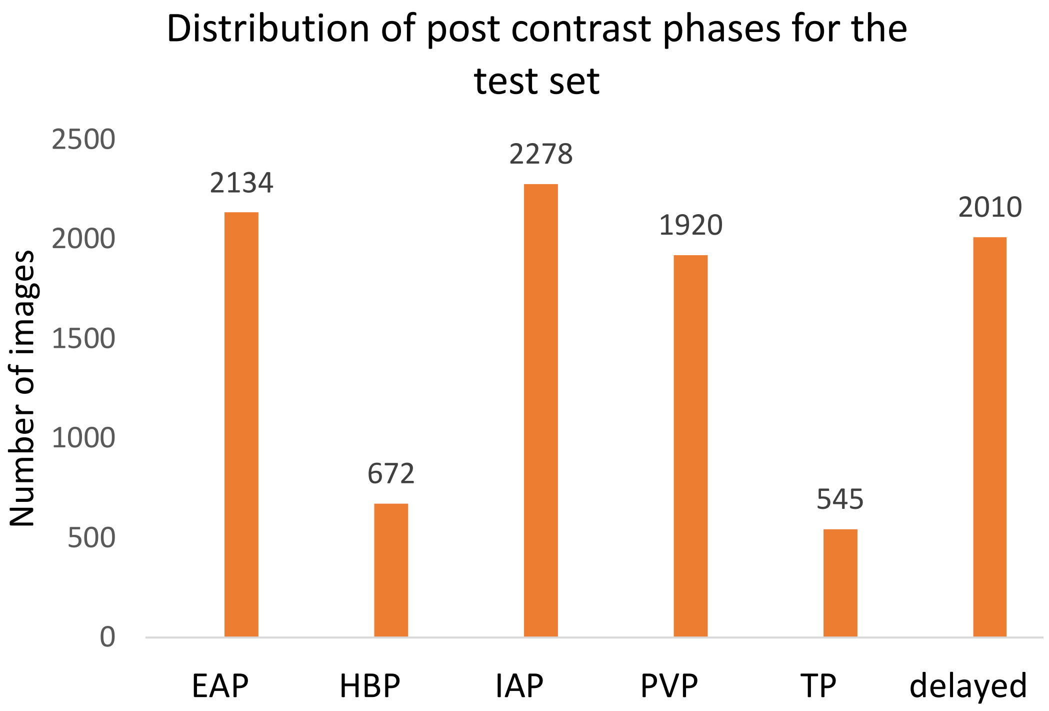


Figure A 13: Number of images per phase in the test set after the split

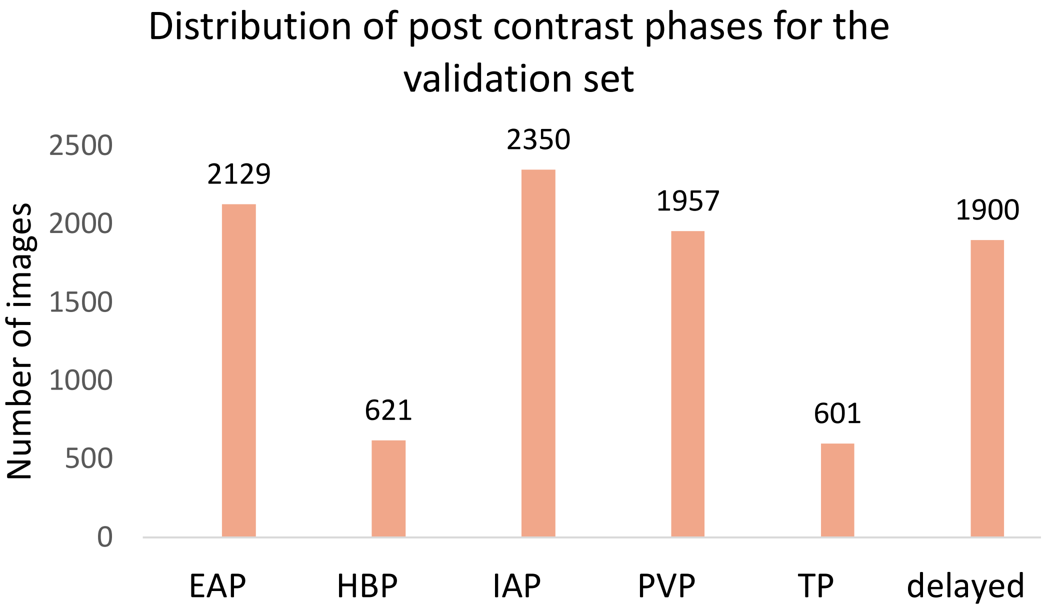


Figure A 14: Number of images per phase in the validation set after the split

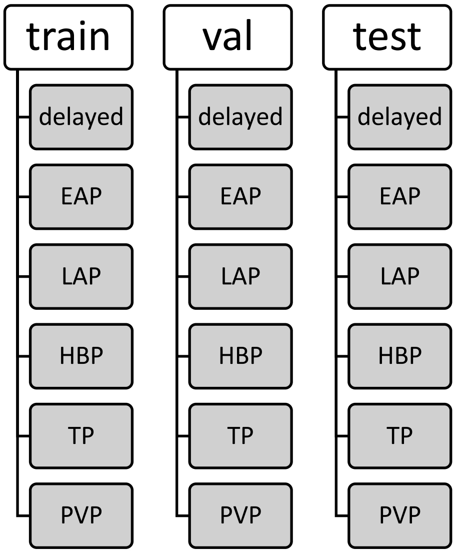


Figure A 15: Phases dataset organization for reading the images by ImageDataGenerator

### A.3.1 Training and Classification Results for Model A on MRI Phases Dataset

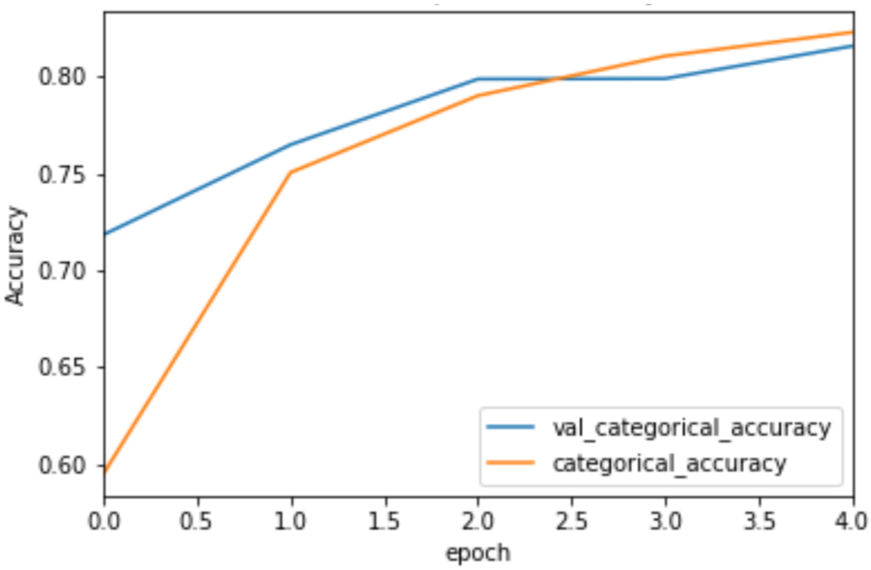


Figure A 16: Validation accuracy for baseline model using phases dataset

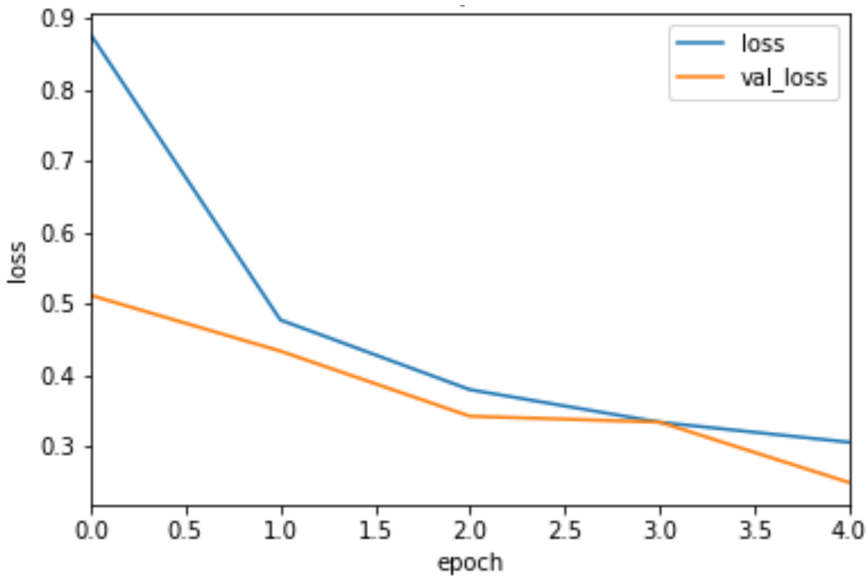
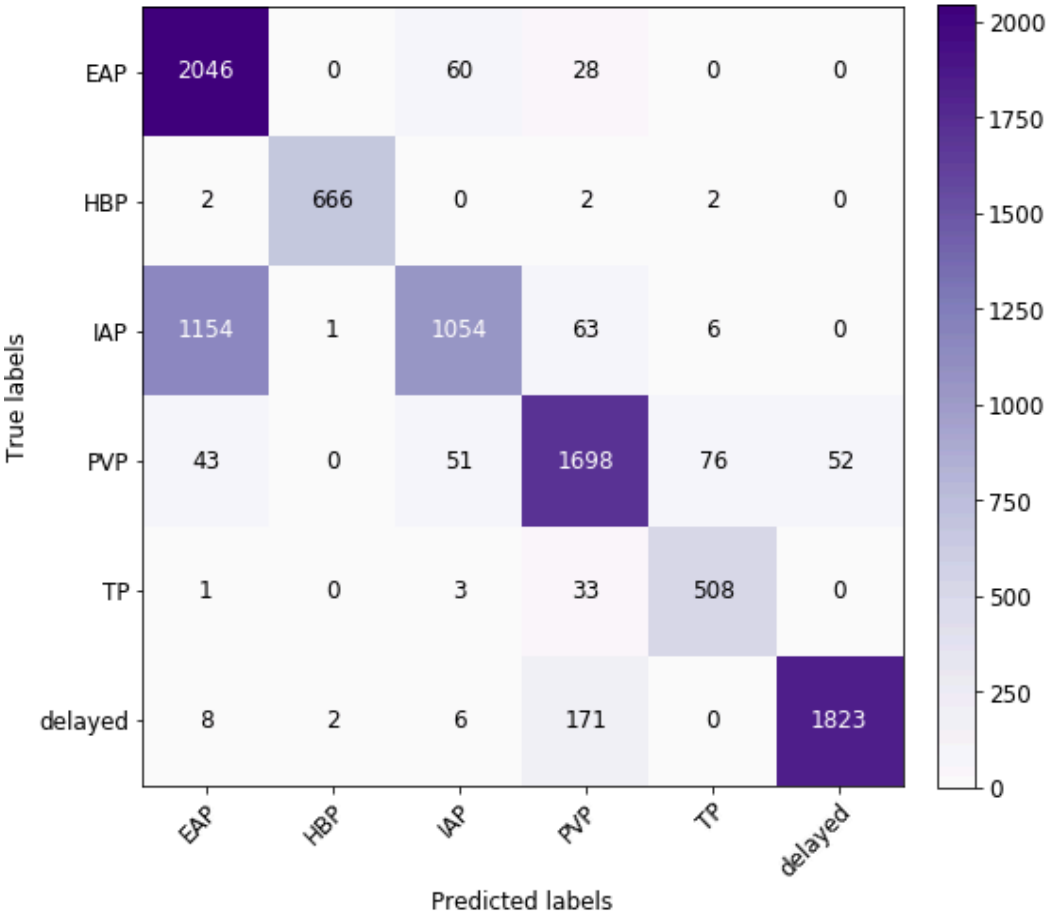


Figure A 17: Validation loss for model using phases dataset



| True labels \ Predicted labels | EAP | HBP | IAP | PVP | TP | delayed |
|---|---|---|---|---|---|---|
| EAP | 2046 | 0 | 60 | 28 | 0 | 0 |
| HBP | 2 | 666 | 0 | 2 | 2 | 0 |
| IAP | 1154 | 1 | 1054 | 63 | 6 | 0 |
| PVP | 43 | 0 | 51 | 1698 | 76 | 52 |
| TP | 1 | 0 | 3 | 33 | 508 | 0 |
| delayed | 8 | 2 | 6 | 171 | 0 | 1823 |

Figure A 18: Confusion matrix for Baseline model using phases dataset. Test set was not shuffled

Table A 5: Classification report for baseline model with phases dataset. Test set was not shuffled.

| | precision | recall | F1-score | support |
|---|---|---|---|---|
| EAP | 0.63 | 0.96 | 0.76 | 2134 |
| HBP | 1 | 0.99 | 0.99 | 672 |
| IAP | 0.9 | 0.46 | 0.61 | 2278 |
| PVP | 0.85 | 0.88 | 0.87 | 1920 |
| TP | 0.86 | 0.93 | 0.89 | 545 |
| delayed | 0.97 | 0.91 | 0.94 | 2010 |
| accuracy | | | 0.82 | 9559 |

| | | | | |
|---|---|---|---|---|
| macro avg | 0.87 | 0.86 | 0.84 | 9559 |
| weighted avg | 0.85 | 0.82 | 0.81 | 9559 |

## A.4 ImageNet Dataset

To test our model’s efficacy, the publicly available ImageNet dataset was selected. A balanced reference dataset (ImageNet) was chosen having 10 classes. ImageNet consists of labelled images belonging to multiple classes and it has been used in deep learning to improve and evaluate model architectures for image classification and object detection as observed in the ImageNet competition.

Modified version of ImageNet data consisting of 10 classes was used as the reference dataset. The dataset was a pre-prepared sub dataset of the original ImageNet dataset. All images contained their respective class labels and the dimensions of the images were 224x224.

The ten classes containing the images were:

1. Cassette player,
2. Chain saw,
3. Church,
4. Garbage truck,
5. Gas pump,

6. French horn,
7. Parachute,
8. Springer,
9. Tench and
10. Golf ball

This dataset was chosen as it represents a multiclass classification similar to the dataset for MRI sequence imaging. Since the dataset is balanced, no geometric augmentation methods have been applied to the training set. Furthermore, their dataset does not contain a validation set as compared to our dataset preparation approach Figure 4 and hence hold-out method (train and test split) was used. The training and test sets were also pre-prepared and 80% of images were observed for the training and 20% for the test set respectively.

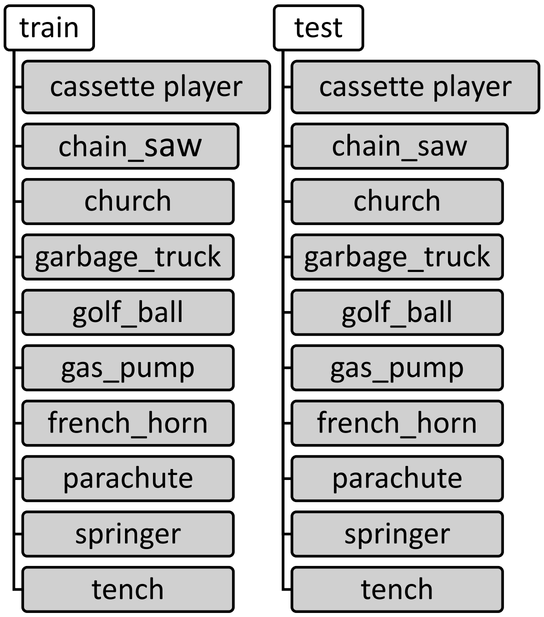


Figure A 19: Dataset structure for reading the ImageNet training and test images using Keras ImageDataGenerator class

Figure A 20: Sample images from the ImageNet dataset

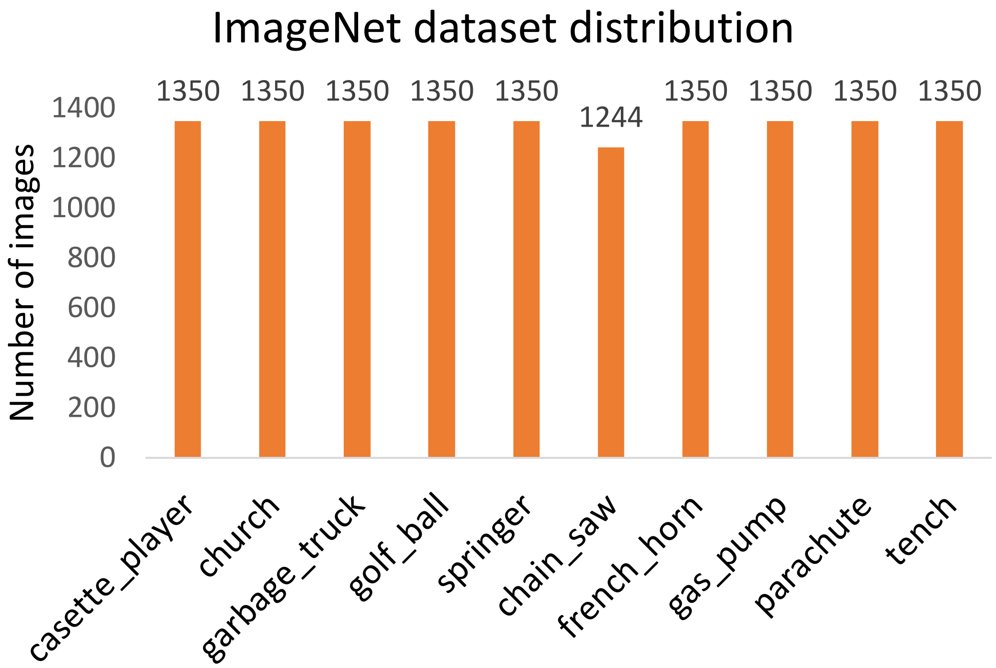


Figure A 21: Balanced distribution of images per class in the reference dataset

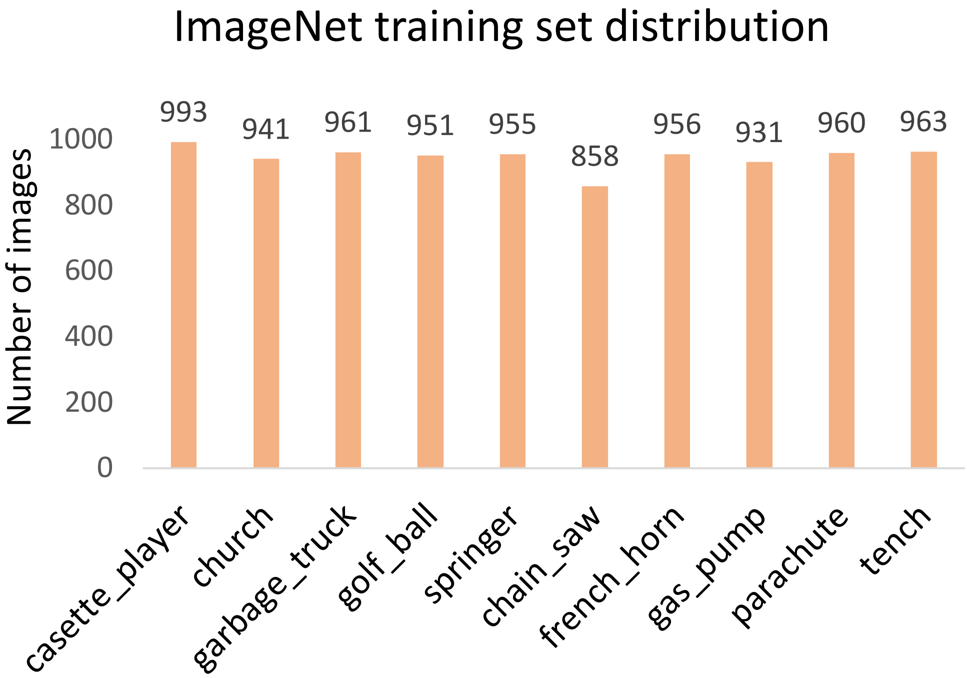


Figure A 22: Training set class distribution in the reference dataset

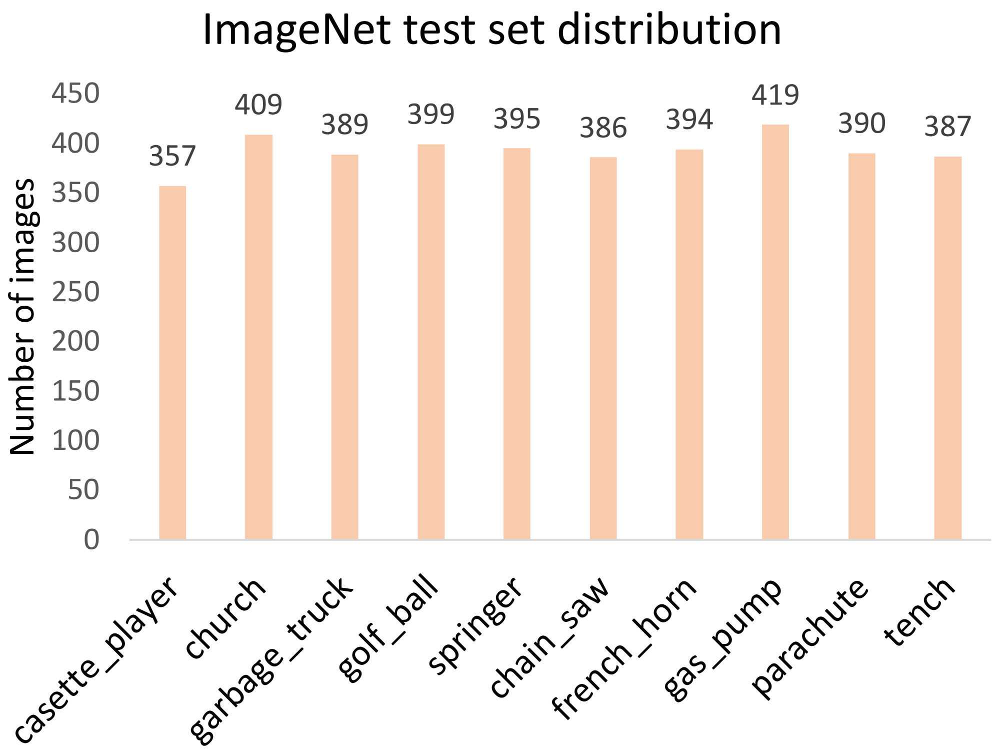


Figure A 23: Test set class distribution for reference dataset

As observed from Figure A 21 to Figure A 23, the distribution of the images per class in training, testing and overall is balanced (same number of images per class).

## 4.4 Model E (Imagenet) - Classification Results

Model E was trained using the Architecture X (Figure 11) on the ImageNet dataset. Classification results of the model on the test have been provided here.

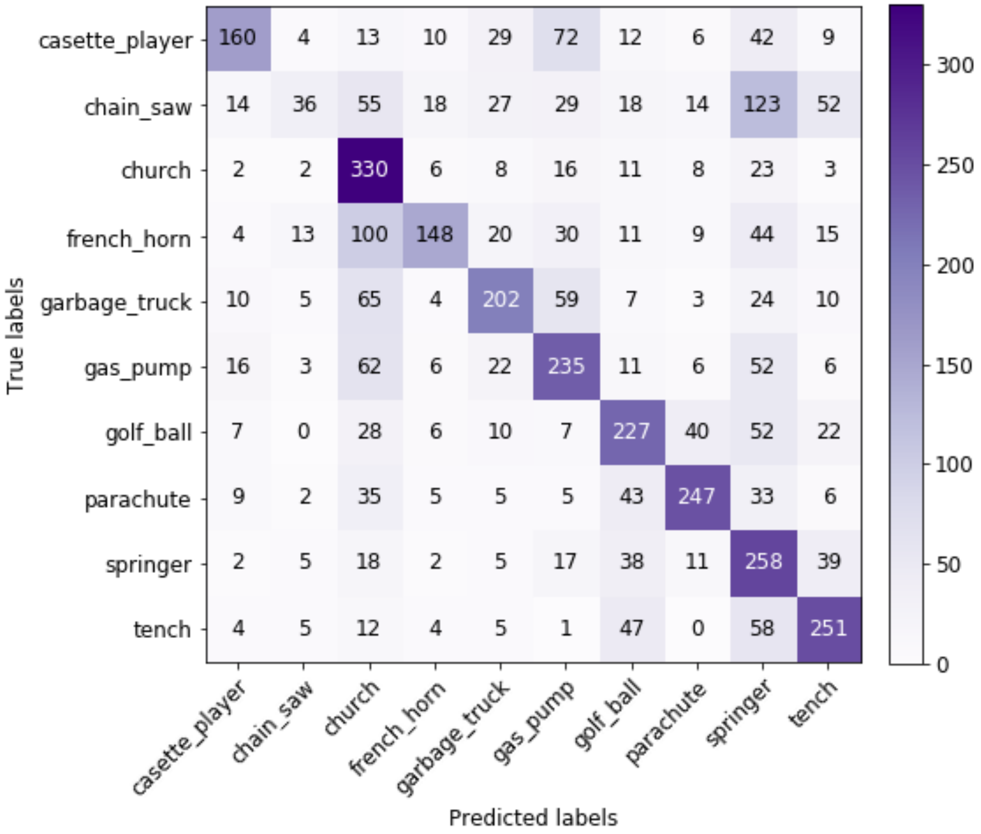


Figure A 24: Confusion matrix results for Model D predictions on the ImageNet test set

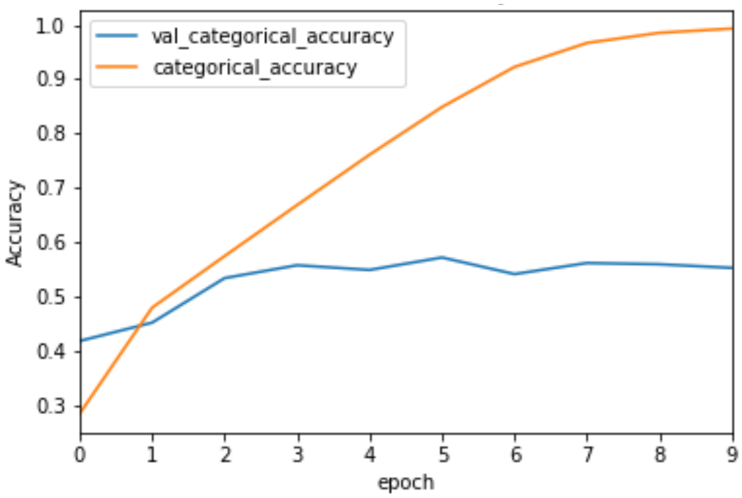


Figure A 25: Validation accuracy plot for Model E (same architecture as baseline Model A) training

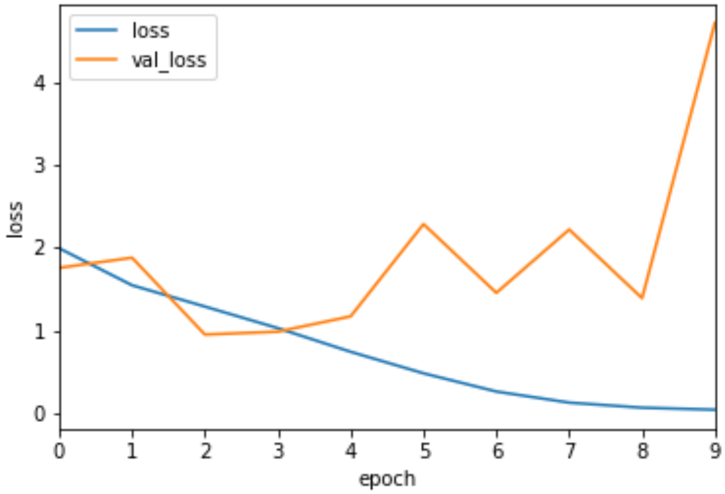


Figure A 26: Training and validation loss for Model E

Table A 6: Classification report for Model E predictions

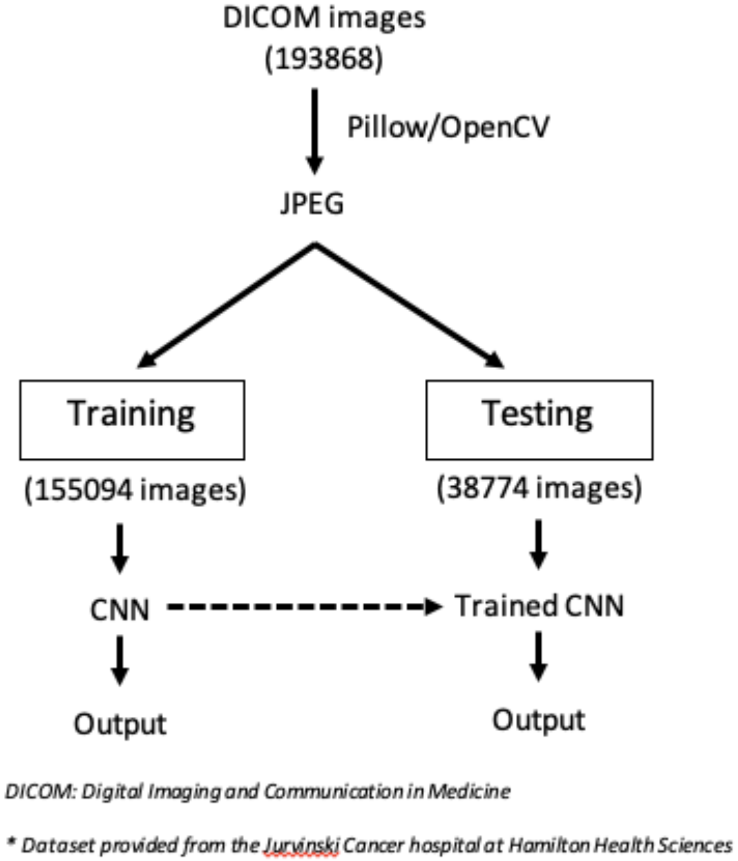
| | precision | recall | F1-score | support |
|---|---|---|---|---|
| casette_player | 0.60 | 0.50 | 0.54 | 357 |
| chain_saw | 0.43 | 0.06 | 0.11 | 409 |
| church | 0.57 | 0.70 | 0.63 | 389 |
| french_horn | 0.82 | 0.32 | 0.46 | 399 |
| garbage_truck | 0.59 | 0.50 | 0.54 | 395 |
| gas_pump | 0.55 | 0.48 | 0.51 | 386 |
| golf_ball | 0.48 | 0.61 | 0.54 | 394 |
| parachute | 0.68 | 0.66 | 0.67 | 419 |
| springer | 0.30 | 0.66 | 0.41 | 390 |
| tench | 0.55 | 0.67 | 0.61 | 387 |
| accuracy | | | 0.52 | 3925 |
| macro avg | 0.56 | 0.51 | 0.50 | 3925 |
| weighted avg | 0.56 | 0.52 | 0.50 | 3925 |

Furthermore, for the balanced distribution using ImageNet dataset, Model E had low precision and recall scores ( Table A 1). An accuracy of 52% was reported for Model E on the test set predictions. As observed from Figure A 24, there were misclassifications which relate to the poor accuracy score. Additionally, chain_saw class shows the lowest recall (0.06) and precision (0.43) scores. Only 36 of chain_saw images were predicted correctly with 13 incorrectly predicted as belonging to the french_horn class.

Using the balanced distribution of images per class in the ImageNet dataset, Model E showed an accuracy of 52%. Model E had similar architecture as that of Model A and after training on the balanced training set it was found that the model does suffers from overfitting. This was evident from the classification results from Figure A 25 and Figure A 26. This could have been due to the model being not able to distinguish certain features that could have overlapped among the classes. For example, the church images had certain features of trees and grass that could have overlapped with those of tench or garbage truck images.

## A.5 Model Training Using MRI Sequence Images in JPEG Format

This section provides method workflow for using JPEG image format for training. It was found that JPEG conversion resulted in loss of features from MRI sequences Figure A 32



*DICOM: Digital Imaging and Communication in Medicine*

** Dataset provided from the Jurvinski Cancer hospital at Hamilton Health Sciences*

Figure A 27: Hold-out method workflow for conversion of DICOM images to JPEG format.

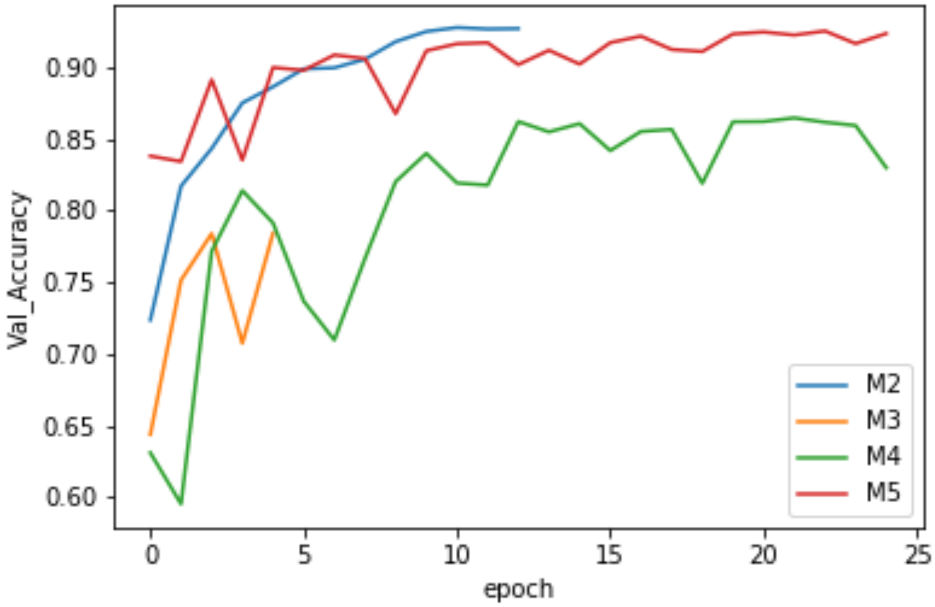


Figure A 28: Validation accuracy for models run using JPEG image format. M5 and M4 used the same architecture as Model A

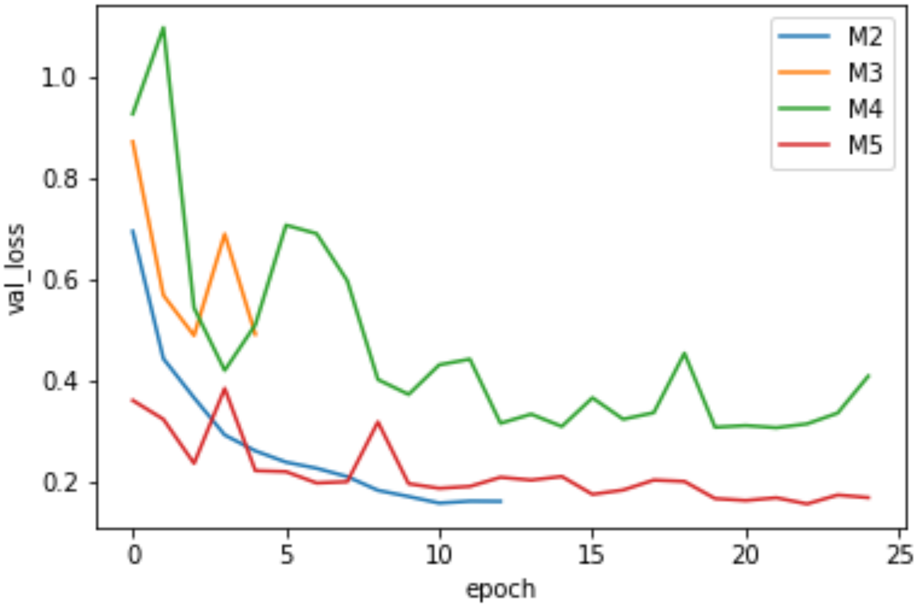
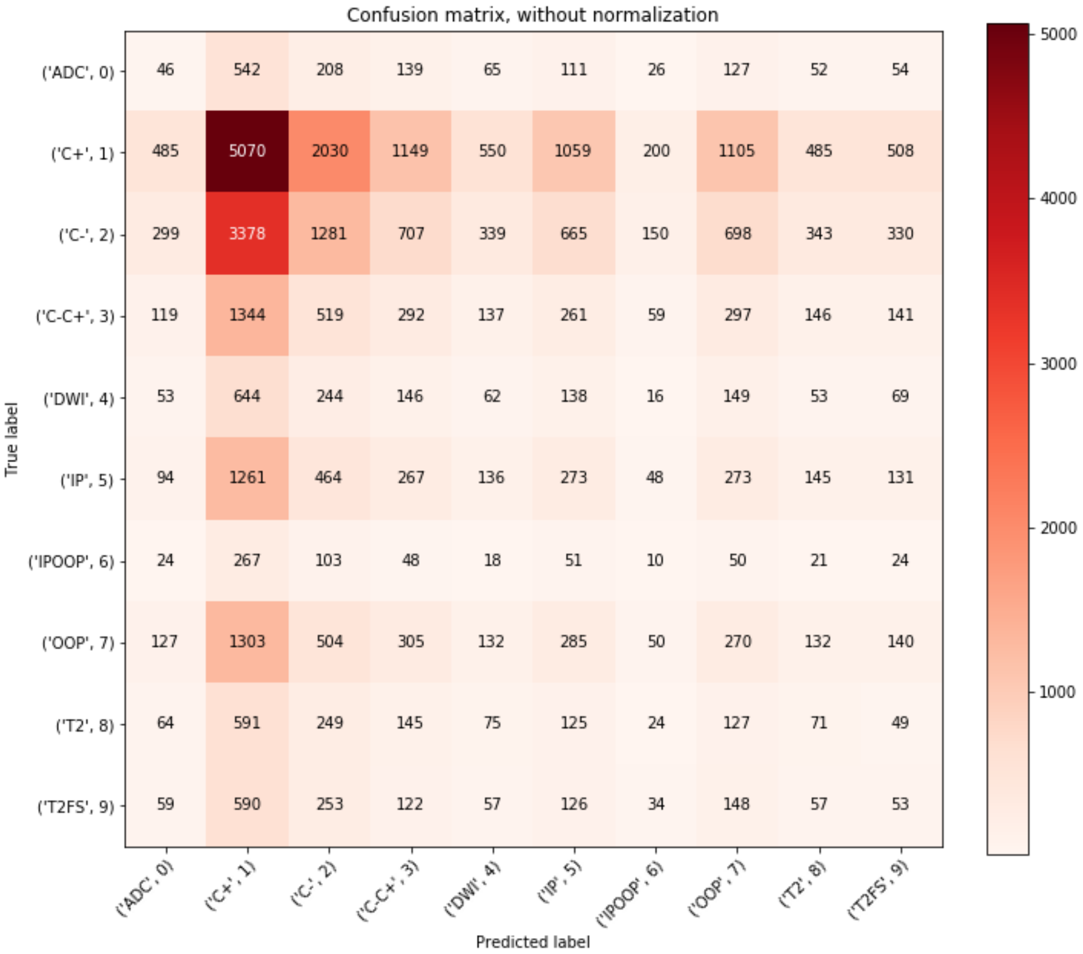


Figure A 29: Validation loss for M2-M5 trained using JPEG image format. M5 and M4 used the same architecture as Model A



| | ('ADC', 0) | ('C+', 1) | ('C-', 2) | ('C-C+', 3) | ('DWI', 4) | ('IP', 5) | ('IPOOP', 6) | ('OOP', 7) | ('T2', 8) | ('T2FS', 9) |
|---|---|---|---|---|---|---|---|---|---|---|
| ('ADC', 0) | 46 | 542 | 208 | 139 | 65 | 111 | 26 | 127 | 52 | 54 |
| ('C+', 1) | 485 | 5070 | 2030 | 1149 | 550 | 1059 | 200 | 1105 | 485 | 508 |
| ('C-', 2) | 299 | 3378 | 1281 | 707 | 339 | 665 | 150 | 698 | 343 | 330 |
| ('C-C+', 3) | 119 | 1344 | 519 | 292 | 137 | 261 | 59 | 297 | 146 | 141 |
| ('DWI', 4) | 53 | 644 | 244 | 146 | 62 | 138 | 16 | 149 | 53 | 69 |
| ('IP', 5) | 94 | 1261 | 464 | 267 | 136 | 273 | 48 | 273 | 145 | 131 |
| ('IPOOP', 6) | 24 | 267 | 103 | 48 | 18 | 51 | 10 | 50 | 21 | 24 |
| ('OOP', 7) | 127 | 1303 | 504 | 305 | 132 | 285 | 50 | 270 | 132 | 140 |
| ('T2', 8) | 64 | 591 | 249 | 145 | 75 | 125 | 24 | 127 | 71 | 49 |
| ('T2FS', 9) | 59 | 590 | 253 | 122 | 57 | 126 | 34 | 148 | 57 | 53 |

Figure A 30: Confusion matrix result for the M5 model. M5 used the augmented JPEG training set

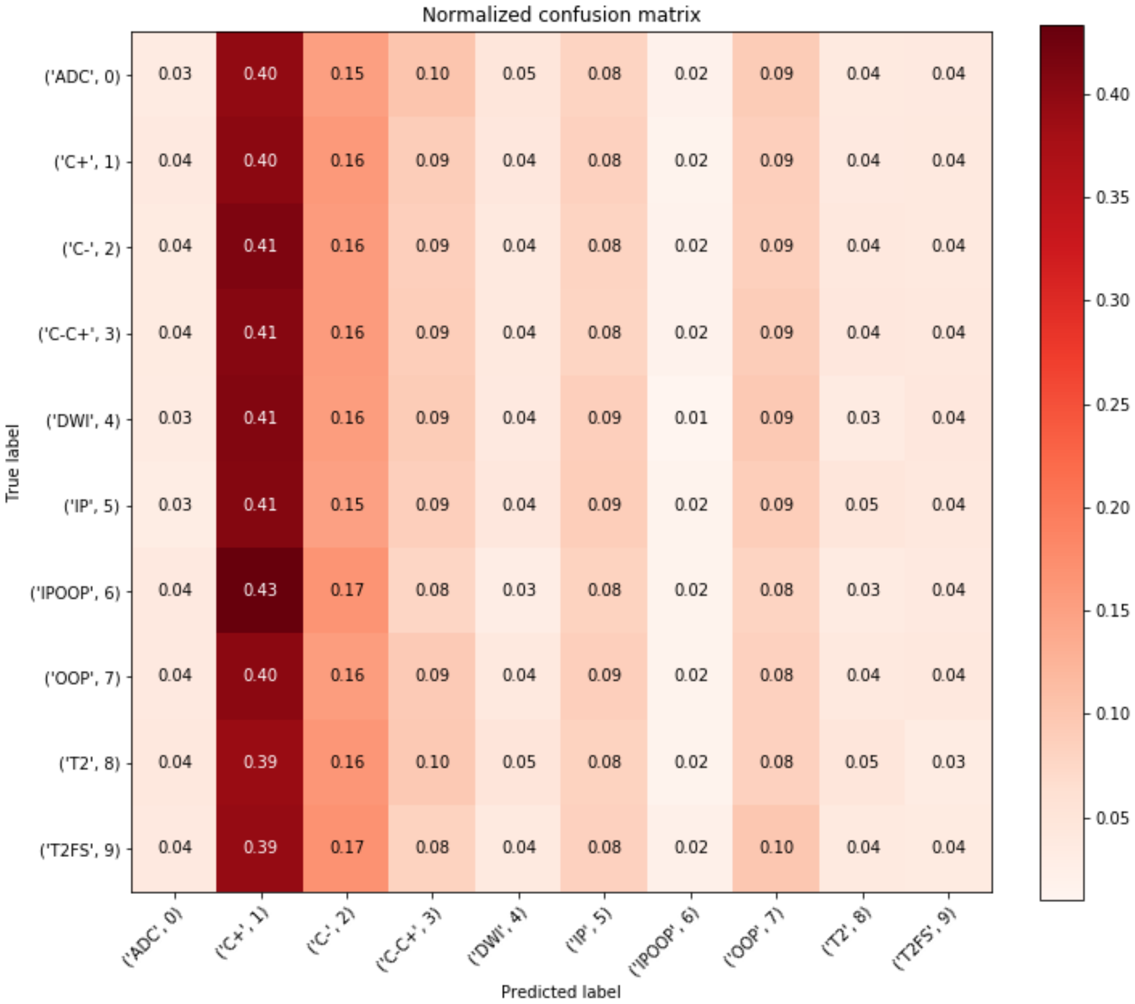


Figure A 31: Normalized confusion matrix result for the M5 model. M5 used the augmented JPEG training set

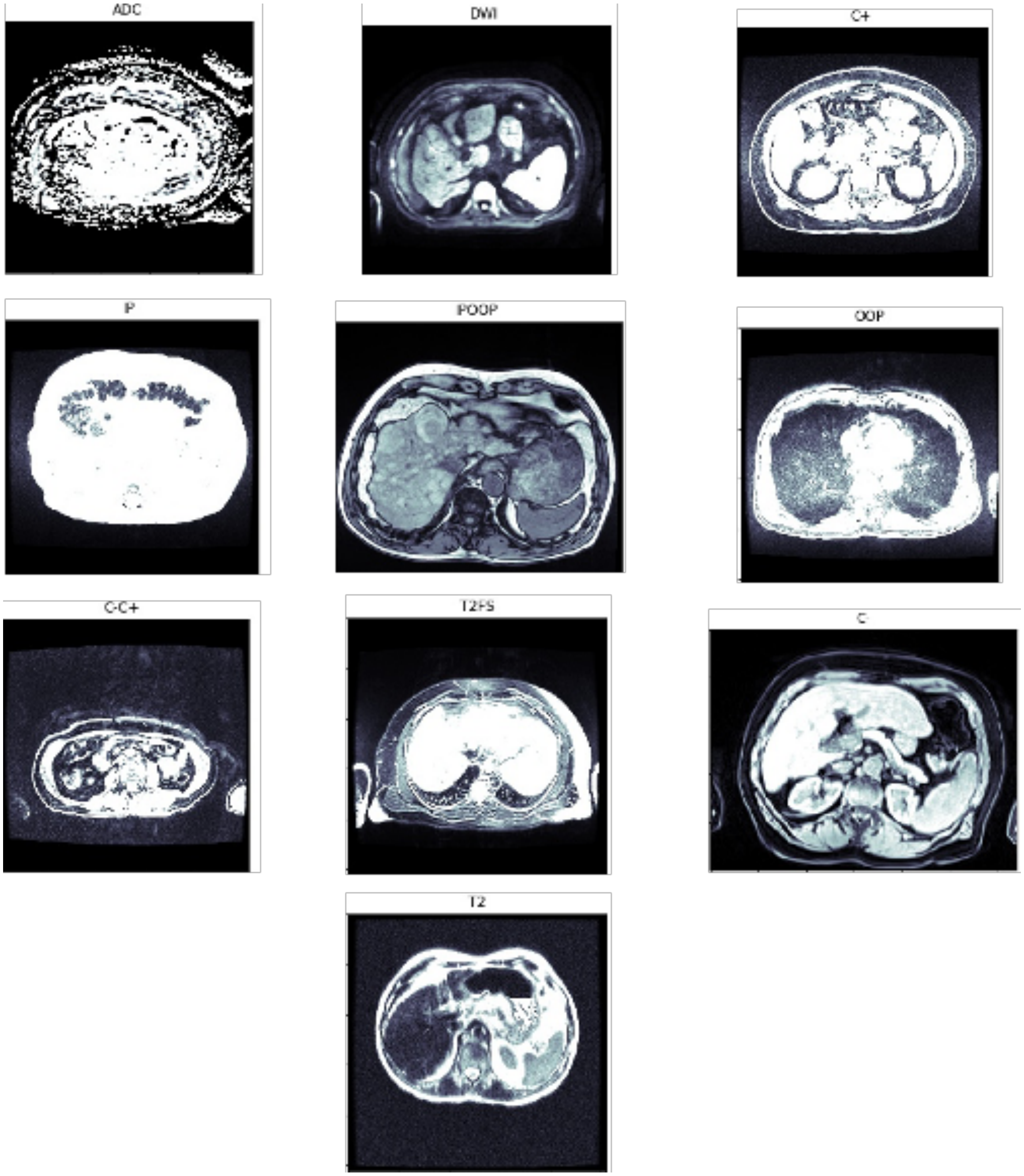


Figure A 32: Sequence dataset JPEG images converted from DICOM image format using OpenCV