# APPLYING CONTINUAL LEARNING STRATEGIES FOR CLASSIFICATION OF CERVICAL CELLS

by

Gökhan Akgün

Submitted to Graduate School of Natural and Applied Sciences

in Partial Fulfillment of the Requirements

for the Degree of Master of Science in

Data Science

Yeditepe University

2022

APPLYING CONTINUAL LEARNING STRATEGIES FOR CLASSIFICATION OF CERVICAL CELLS

APPROVED BY:

Assoc. Prof. Dr. Dionysis Goularas ……
(Thesis Supervisor)
(Yeditepe University)

Prof. Dr. Cem Unsalan ……
(Marmara University)

Prof. Dr. Emin Erkan Korkmaz ……
(Yeditepe University)

DATE OF APPROVAL: …. / …. / 20….

I hereby declare that this thesis is my own work and that all information in this thesis has been obtained and presented in accordance with academic rules and ethical conduct. I have fully cited and referenced all material and results as required by these rules and conduct, and this thesis study does not contain any plagiarism. If any material used in the thesis requires copyright, the necessary permissions have been obtained. No material from this thesis has been used for the award of another degree.

I accept all kinds of legal liability that may arise in cases contrary to these situations.

Name, Last Name Gökhan Akgün

Signature ……

# ACKNOWLEDGEMENTS

I am extremely grateful to my Professor Dionysis Goularas for his assistance and support. Pursuing my thesis under his supervision has been an enriching experience that has provided an endless source of learning.

I thank my close friends Yusufcan Semerci and Çağrı Yeşil for their great support.

Finally, I would like to express my gratitude to my family for their unending love and support, which makes everything more beautiful.

# ABSTRACT


## APPLYING CONTINUAL LEARNING STRATEGIES FOR CLASSIFICATION OF CERVICAL CELLS

Medical imaging plays an essential role in clinical diagnosis and therapy planning. Machine learning is gaining popularity because it captures illness and therapy response characteristics. The continual improvement of image collecting technology and diagnostic methods, the diversity of scanners and developing imaging protocols, and field shifts restrict the usefulness of machine learning as predicted accuracy on new data degrades or models become out-of-date. An example for such shifts is pap smear imaging. This thesis presents applications of continual learning strategies for tackling models' outdating due to the domain shifts in pap smear images. Two popular convolutiunal neural network (CNN) architectures (InceptionV3 and ResNet50), and a custom 2-layer CNN are trained with three pap smear benchmark datasets (Sipakmed, Herlev, and CRIC). Several continual learning strategies are employed when training the models. Finally, the models for each strategy are evaluated, and the comparative results are presented.

# ÖZET

## SERVİKS HÜCRELERİNİN SINIFLANDIRILMASINDA SÜREKLİ ÖĞRENME STRATEJİLERİNİN UYGULANMASI

Tıbbi görüntüleme, klinik tanı ve tedavi planlamasında önemli bir rol oynar. Makine öğrenimi ise, hastalık ve tedaviye yanıt özelliklerini yakaladığı için popülerlik kazanmaktadır. Görüntü elde etme teknolojisinin ve tanı prosedürlerinin sürekli ilerlemesi, tarayıcıların çeşitliliği ve gelişen görüntüleme protokolleri, alan kaymaları nedeniyle yeni veriler üzerindeki tahmin doğruluğu bozulduğundan veya modeller eskidiğinden, makine öğreniminin faydası sınırlanır. Bu tür alan kaymalarının bir örneği de pap smear görüntülerindedir. Bu tezde, pap smear görüntülerindeki alan kaymaları nedeniyle modellerin güncelliğini yitirmesiyle mücadele etmek için sürekli öğrenme stratejilerinin uygulamaları sunulmuştur. İki popüler evrişimli sinir ağı mimarisi (InceptionV3 ve ResNet50) ve 2 katmanlı evrişimli sinir ağı modeli, üç pap smear kıyaslama veri seti (Sipakmed, Herlev ve CRIC) ile eğitilmiştir. Modeller eğitilirken çeşitli sürekli öğrenme stratejileri kullanılmıştır. Son olarak, her bir strateji için modeller değerlendirilerek karşılaştırmalı sonuçlar sunulmuştur.

# TABLE OF CONTENTS

ACKNOWLEDGEMENTS ..... iv
ABSTRACT ..... v
ÖZET ..... vi
TABLE OF CONTENTS ..... vii
LIST OF FIGURES ..... x
LIST OF TABLES ..... xii
LIST OF SYMBOLS/ABBREVIATIONS ..... xiii
1. INTRODUCTION ..... 1
2. BACKGROUND ..... 5
3. THEORETICAL BACKGROUND ..... 11
3.1. CERVICAL SCREENING ..... 11
3.2. IMAGE AUGMENTATION ..... 11
3.2.1. Affine Transformations ..... 11
3.2.2. Contrast Limited Adaptive Histogram Equalization ..... 12
3.2.3. Edge Detection ..... 12
3.2.4. Sharpening ..... 12
3.2.5. Canny Filter ..... 12
3.2.6. Channelwise Transformations ..... 13
3.2.7. Applying Noise ..... 13
3.2.8. Photometric Transformations ..... 13
3.2.9. Contrast Adaptation ..... 13
3.3. LEARNING ALGORITHMS ..... 14
3.3.1. Convolution ..... 14
3.3.2. Pooling ..... 14
3.3.3. Artificial Neural Networks ..... 16
3.3.4. Convolutional Neural Networks ..... 17
3.4. ADVANCED LEARNING TECHNOLOGIES ..... 17
3.4.1. CNN Architectures ..... 17
3.4.1.1. InceptionV3 ..... 17
3.4.1.2. ResNet50 ..... 21

3.4.2. Continual Learning ..... 22
3.4.2.1. Naive Strategy ..... 22
3.4.2.2. Rehearsal Strategy ..... 23
3.4.2.3. Elastic Weight Consolidation ..... 23
4. METHODOLOGY ..... 25
4.1. DATASETS ..... 25
4.1.1. Sipakmed Dataset ..... 25
4.1.2. Herlev Dataset ..... 26
4.1.3. CRIC Dataset ..... 27
4.2. DATASET PREPARATION ..... 29
4.3. MODELS ..... 30
4.3.1. Pre-trained Models ..... 30
4.3.2. Custom CNN ..... 30
4.4. CONTINUAL LEARNING STRATEGIES ..... 31
4.4.1. Naive Strategy ..... 31
4.4.2. Rehearsal Strategy ..... 32
4.4.3. EWC Strategy ..... 32
4.4.4. Rehearsal EWC Strategy ..... 32
4.5. EVALUATION METHOD ..... 32
4.5.1. Confusion Matrix ..... 33
4.5.2. Accuracy ..... 33
4.5.3. Specificity ..... 33
4.5.4. Recall (Sensitivity) ..... 34
4.5.5. Precision ..... 34
4.5.6. F1 Score ..... 34
4.5.7. McNemar's Test ..... 34
5. IMPLEMENTATION ..... 36
6. RESULTS ..... 37
6.1. NETWORK MODEL ANALYSIS ..... 37
6.1.1. Naive Strategy ..... 37
6.1.2. Rehearsal Strategy ..... 39
6.1.3. EWC Strategy ..... 40

6.1.4. Rehearsal EWC Strategy ..... 42
6.2. STRATEGY ANALYSIS ..... 43
6.2.1. Resnet50 Model ..... 43
6.2.2. InceptionV3 Model ..... 45
6.2.3. 2-Layer CNN Model ..... 46
7. CONCLUSION ..... 49
REFERENCES ..... 50

# LIST OF FIGURES

Figure 1.1. The general overview of the proposed approach ..... 4

Figure 2.1. Overview of the suggested method for categorizing cervical cell images using convolutional neural networks and transfer learning ..... 5

Figure 2.2. CNN framework's overall flowchart for a 7-class classification problem ..... 6

Figure 2.3. The proposed method's structure ..... 6

Figure 2.4. The suggested hybrid deep feature fusion network's framework ..... 7

Figure 2.5. The suggested fuzzy rank-based ensemble of CNN models' overall structure, which is utilized to classify cervical cytology ..... 8

Figure 2.6. The proposed deep model and cervical cancer detection pipelines ..... 9

Figure 2.7. Various model training and adaptation strategies ..... 10

Figure 3.1. Average Pooling ..... 15

Figure 3.2. Max Pooling ..... 16

Figure 3.3. Stem Block ..... 18

Figure 3.4. Inception Block (Type 1) ..... 18

Figure 3.5. Inception Block (Type 2) ..... 19

Figure 3.6. Inception Block (Type 3) ..... 19

Figure 3.7. Reduction Block (Type 1) ..... 20

Figure 3.8. Reduction Block (Type 2) ..... 20

Figure 3.9. Auxiliary Classifier ..... 20

Figure 3.10. InceptionV3 Overall Architecture ..... 21

Figure 3.11. Convolutional Block ..... 21

Figure 3.12. Identity Block ..... 22

Figure 3.13. ResNet50 Overall Architecture ..... 22

Figure 3.14. When practicing task B, EWC makes sure task A is retained. EWC handles task B without a substantial loss on task A. In a schematic representation of parameter space, training trajectories are shown, and parameter regions that result in successful completion of tasks A and B are highlighted. ..... 23

Figure 4.1. Sample images for each 5 category of Sipakmed database ..... 25

Figure 4.2. Sample images for each 7 category of Herlev database ..... 27

Figure 4.3. Sample images for each 6 category of CRIC database ..... 28

Figure 4.4. The structure of proposed models ..... 31

Figure 6.1. The comparison of Resnet50, InceptionV3, and 2-layer CNN models with naive strategy ..... 38

Figure 6.2. The comparison of Resnet50, InceptionV3, and 2-layer CNN models with rehearsal strategy ..... 40

Figure 6.3. The comparison of Resnet50, InceptionV3, and 2-layer CNN models with EWC strategy ..... 41

Figure 6.4. The comparison of Resnet50, InceptionV3, and 2-layer CNN models with rehearsal-EWC strategy ..... 43

Figure 6.5. The comparison of naive, rehearsal, EWC, and rehearsal-EWC strategies with Resnet50 model ..... 44

Figure 6.6. The comparison of naive, rehearsal, EWC, and rehearsal-EWC strategies with InceptionV3 model ..... 46

Figure 6.7. The comparison of naive, rehearsal, EWC, and rehearsal-EWC strategies with 2-layer CNN model ..... 48

# LIST OF TABLES

Table 4.1. Sipakmed Database....26

Table 4.2. Herlev Database ....26

Table 4.3. CRIC Database ....28

Table 4.4. The Sipakmed dataset's experimental data settings. ....29

Table 4.5. The Herlev dataset's experimental data settings. ....29

Table 4.6. The CRIC dataset's experimental data settings. ....30

Table 4.7. Binary classification confusion matrix ....33

Table 4.8. Contingency table for McNemar's test ....35

Table 6.1. Comparative performance metrics of Resnet50, InceptionV3 and 2-layer CNN model with Naive strategy ....37

Table 6.2. Comparative performance metrics of Resnet50, InceptionV3 and 2-layer CNN model with Rehearsal strategy ....39

Table 6.3. Comparative performance metrics of Resnet50, InceptionV3 and 2-layer CNN model with EWC strategy ....41

Table 6.4. Comparative performance metrics of Resnet50, InceptionV3 and 2-layer CNN model with Rehearsal EWC strategy ....42

Table 6.5. Comparative performance metrics of Naive, Rehearsal, EWC, and Rehearsal EWC strategies with Resnet50 model ....44

Table 6.6. Comparative performance metrics of Naive, Rehearsal, EWC, and Rehearsal EWC strategies with InceptionV3 model ....45

Table 6.7. Comparative performance metrics of Naive, Rehearsal, EWC, and Rehearsal EWC strategies with 2-layer CNN model ....47

# LIST OF SYMBOLS/ABBREVIATIONS

| | |
|---|---|
| CAD | Computer aided diagnosis |
| CNN | Convolutional neural network |
| CRIC | Center for Recognition and Inspection of Cells |
| EWC | Elastic weight consolidation |
| HDFF | Hybrid deep feature fusion |
| DL | Deep learning |
| LF | Late fusion |
| LBC | Liquid based cytology |
| ReLU | Rectified linear unit |
| AHE | Adaptive histogram equalization |
| CLAHE | Contrast limited adaptive histogram equalization |
| ANN | Artificial neural network |
| NILM | Negative for intraepithelial lesion or malignancy |
| ASC-US | Atypical squamous cells of unclear significance |
| LSIL | Low-grade squamous intraepithelial lesion |
| ASC-H | Atypical squamous cells, cannot exclude a high-grade lesion |
| HSIL | High-grade squamous intraepithelial lesion |
| SCC | Squamous cell carcinoma |
| CL | Continual learning |
| TP | True positive |
| TN | True negative |
| FN | False negative |
| FP | False positive |
| SGD | Stochastic gradient descent |

# 1. INTRODUCTION

Cervical cancer is the fourth most frequent malignancy in women worldwide. Also, it is the fourth most frequent cancer in terms of causing death [1]. A cytopathological screening method called Pap smear is being applied for the diagnosis of lesions that are precursors of cervical cancer. Cell samples taken from the person are being examined under a microscope and diagnosed after going through fixation, staining, and smear processes. Early diagnosis of cervical cancer allows the disease to be prevented and treated [2].

It takes a lot of effort and is prone to mistakes in examining pap smear slides. In order to minimize errors, cytotechnologists work with high concentration. It is recommended that cytotechnologists should not work more than 7 hours and should not analyze more than 70 slides per day [3]. It has always been a challenge to find qualified cytotechnologists to screen and diagnose Pap smear slides, which is why creating an automated system is a top priority [4]. By supplying data for Computer Aided Diagnosis (CAD), digital pathology plays a key role in disease diagnosis when compared to manual assessment.

The key drawback of the above method is that while the characteristics retrieved for classification are often created by hand, the effectiveness of the classification step cannot be guaranteed. While offering automatic learning of features based on certain objective functions like detection, segmentation, and classification, the deep learning methodology is a type of learning method that may directly analyze raw data [5]. Deep learning techniques are being used in natural language processing, computer vision, image understanding, computational biology, and medical imaging. They are making substantial contributions to the field of artificial intelligence as well as human performance and their applications in a wide range of fields including medicine and engineering [6,7].

One of the main areas of study in medical image analysis is image classification. Multiple photos are inputted into the computer vision process of "image classification" which then categorizes the images based on their visual content.

In general, building a neural network from scratch necessitates a significant quantity of data and processing capacity, which also extends training timeframes. Transfer learning, which employs the previously trained model from one job to tackle a second task, is the answer to all

of these issues [8]. Since organizing high-quality imaging datasets is difficult and the datasets are often tiny, transfer learning is extremely beneficial in the field of medical imaging.

Diagnostic procedures, imaging-based disease indicators, and even clinical imaging technologies are all evolving. Instead, they have to adapt to a constantly changing environment so that deep learning algorithms can continue to be useful. Deep learning models are now trained just once, resulting in an outstanding performance on images that is consistent with their training experience. However, they have a short lifespan because technological developments cause them to become dated [9]. It greatly limits their usefulness and uptake in clinical practice since they are unable to adjust to fresh data that differs in some ways from the training data.

When the distribution of the training data and the distribution of the data during model inference diverge, a dataset shift occurs [10,11]. Domain shift, a particular kind of such shift, could happen as a result of advancements made in scanner technology. Medical imaging research and clinical practice typically use data from various scanners, generations of scanners, manufacturers, or imaging methods. To properly adapt deployed deep learning models to changing environments, approaches that take these domain shifts into account must be developed and advanced.

The basic goal of continuous learning is to develop a model's capacity to handle new domains using machine learning techniques [12,13]. Catastrophic forgetting is a significant undesirable consequence that is mitigated by ongoing learning techniques [14]. This occurs when updating a model to learn new tasks results in a decline in performance on earlier tasks.

Although several studies investigate continual learning approaches on popular benchmarks such as MNIST, CIFAR, etc., the literature lacks comparable methods of continual learning in the medical domain. To the best of my knowledge, the investigation of continual learning strategies on pap smear data does not exist.

This thesis proposes a comparative analysis of various continual learning strategies on pap smear images. In order to achieve this, a benchmark of pap smear datasets is created by combining three different pap smear datasets: Sipakmed, Herlev, and Center for Recognition and Inspection of Cells (CRIC) datasets. Then, two different Convolutional Neural Network

(CNN) architectures are determined as a feature extractor of the image classifier model. A classifier is integrated to the feature extractors to predict the pap smear images. A 2-layer custom CNN network has been built in addition to these two popular architectures. Finally, four different continual learning strategies are implemented, and these three models are trained according to strategies.

The naïve technique sequentially trains the models with the Sipakmed, Herlev, and CRIC datasets. In the rehearsal technique, the subsequent dataset is integrated with the prior one after each training phase. Then, like in the naïve method, the models in the Elastic Weight Consolidation (EWC) strategy are trained consecutively. However, after each training phase, the significance of the learned model parameters is determined and utilized to update the weights in the following training phase. Finally, the rehearsal and EWC strategies are combined. Figure 1.1 depicts a high-level summary of the thesis methodology.

The following chapter of this thesis presents related research on the categorization of pap smears and continual learning in the field of medical. Theoretical details concerning cervical screening and the methods employed in the suggested method are presented in Chapter 3. Chapter 4 provides the suggested approaches' methodology. Implementation of experiments are covered in Chapter 5. The outcomes of the studies are presented in Chapter 6. Finally, Chapter 7 provides the discussion and conclusion.

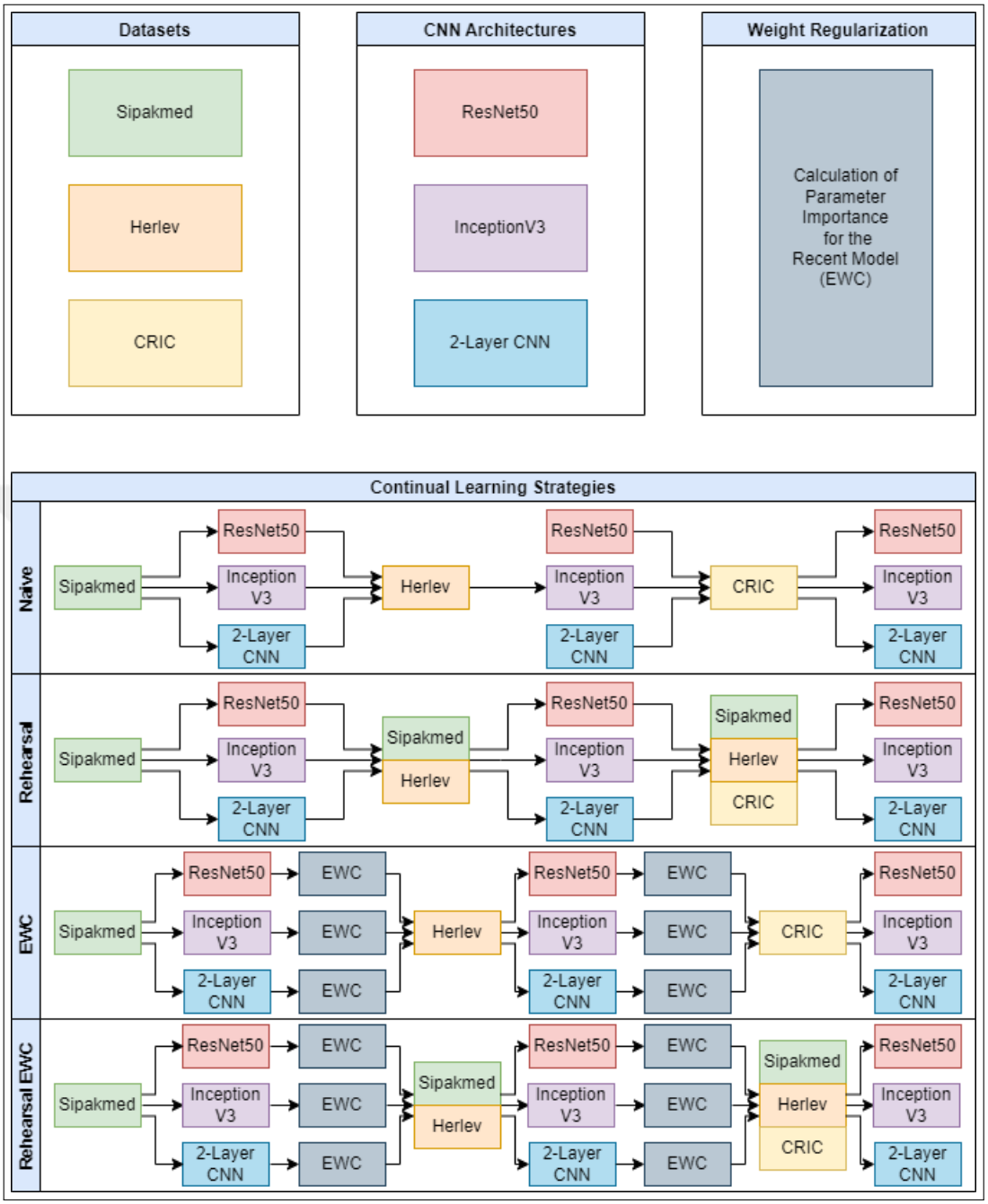



Figure 1.1. The general overview of the proposed approach

# 2. BACKGROUND

Numerous studies have been conducted on the creation of CAD systems over the last few decades, which can assist medical professionals in tracking cervical cancer. From the cytology specification, these systems can automatically identify and categorize the malignant cells [15,16].

To increase the image quality in a standard CAD system, some preprocessing work is first performed using filtering techniques. Then, the segmented nucleus is corrected through postprocessing after the cell nuclei have been extracted using a segmentation method. Following the extraction of features from each nucleus, feature selection is used to choose the feature with the highest level of discrimination. Lastly, a classifier is created to categorize the cell [17].

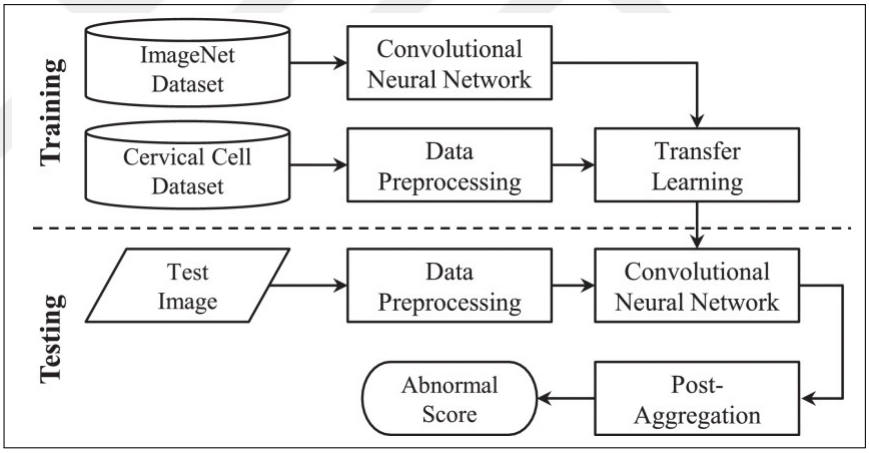



Figure 2.1. Overview of the suggested method for categorizing cervical cell images using convolutional neural networks and transfer learning [18]

[18] proposes the first application of deep learning and transfer learning methods for classifying cervical cells as normal or pathological. The proposed technique was tested using the Herlev and HEMLBC (private) datasets. The preprocessing procedure includes patch extraction and data augmentation. The training and testing steps are illustrated in Figure 2.1 [18]. The accuracy, specificity, sensitivity, and F1 score for the Herlev dataset are 98.3%, 98.3%, 98.2%, and 98.8%, respectively. The accuracy was 98.6%, the specificity was 99%, and the sensitivity was 98.3% on the HEMLBC dataset.

[19] presents a novel cervical cell investigation that connects morphological and

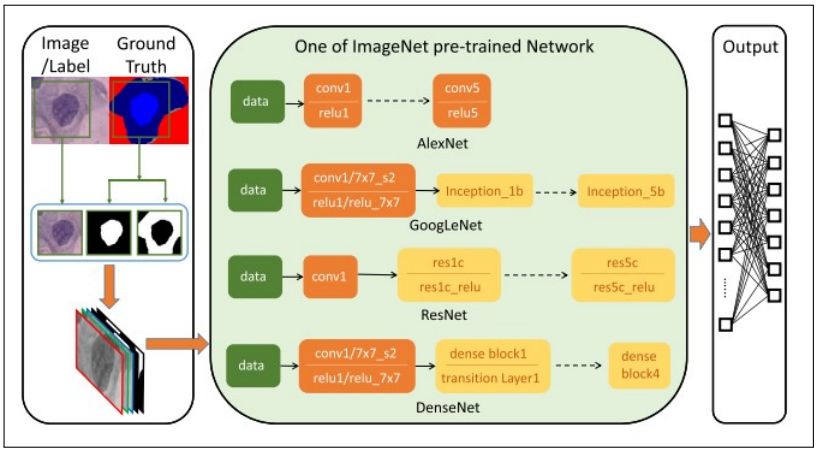



Figure 2.2. CNN framework's overall flowchart for a 7-class classification problem [19].

appearance-based parameters with in-depth aspects. They looked at how CNNs that are morphology and appearance-based together offer better classification accuracy than each one alone. The cervical dataset was used to fine-tune some CNN models (DenseNet, ResNet, GoogLeNet, and AlexNet) that had been pre-trained on the ImageNet dataset as shown in Figure 2.2 [19]. On the Herlev cervical dataset, the suggested approach is assessed. In the preprocessing step, patch and cell morphology extraction, as well as data augmentation, are carried out. GoogLeNet achieves the best classification accuracies among the four CNNs, with 64.5%, 71.3%, and 94.5% for 7-class, 4-class, and 2-class classification tasks, respectively. It should be noted that the nucleus and cytoplasm must first be segmented before using this procedure.

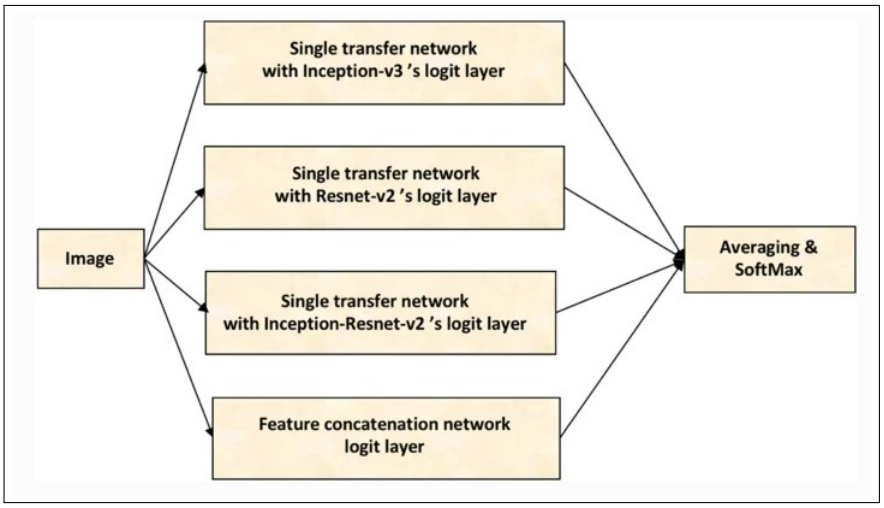



Figure 2.3. The proposed method's structure [20]

In [20], feature concatenation and ensemble techniques for classifying biomedical pictures are proposed by combining several CNNs (InceptionV3, Resnet152, and Inception-Resnet-v2) with varied depths and architectures as shown in Figure 2.3. Three publicly available datasets, including the PAP smear dataset, the 2D Hela dataset, and the Hep-2 cell image dataset, were used to evaluate the performance of the suggested method. It has been observed that a model ensemble rather than a single model produces more trustworthy findings. On the Herlev dataset, an ensemble comprising InceptionV3, Resnet152, Inception-Resnet-v2, and feature concatenation of these three achieved 93.04% accuracy.

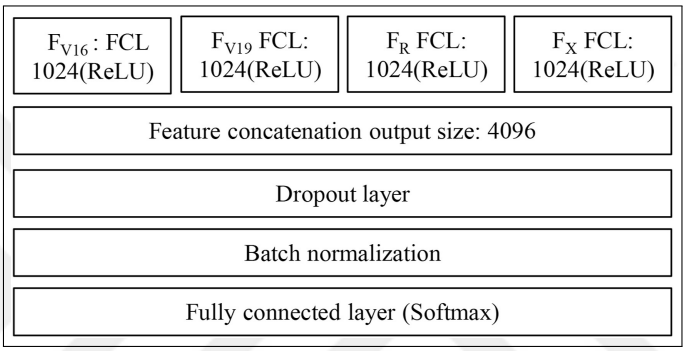



Figure 2.4. The suggested hybrid deep feature fusion network's framework [21].

A Hybrid Deep Feature Fusion (HDFF) technique based on Deep Learning (DL) is proposed in [21] to classify cervical cells accurately. To improve classification performance, the proposed method employs various DL models to capture more potential information. Figure 2.4 illustrates the proposed DeepCervix model's workflow diagram [21].The suggested HDFF approach has been evaluated on the publicly available SIPaKMeD dataset and its performance has been compared to that of basic DL models and the late fusion (LF) method. This method achieves an accuracy of 98.32% for 2-class classification on the Herlev dataset and 90.32% for 7-class classifications on the same dataset. On the other hand, state-of-the-art classification accuracy is obtained for the SIPaKMeD dataset for 5-class, 3-class, and 2-class classification, 99.14%, 99.38%, and 99.85%, respectively.

In [22], an ensemble-based classification model for the categorization of Pap-stained single cells and whole-slide images is developed using three CNN architectures, namely DenseNet-169, InceptionV3, and Xception pre-trained on ImageNet dataset as illustrated in Figure 2.5. The suggested ensemble technique considers two non-linear functions on the decision scores produced by base learners and employs a fuzzy rank-based fusion of

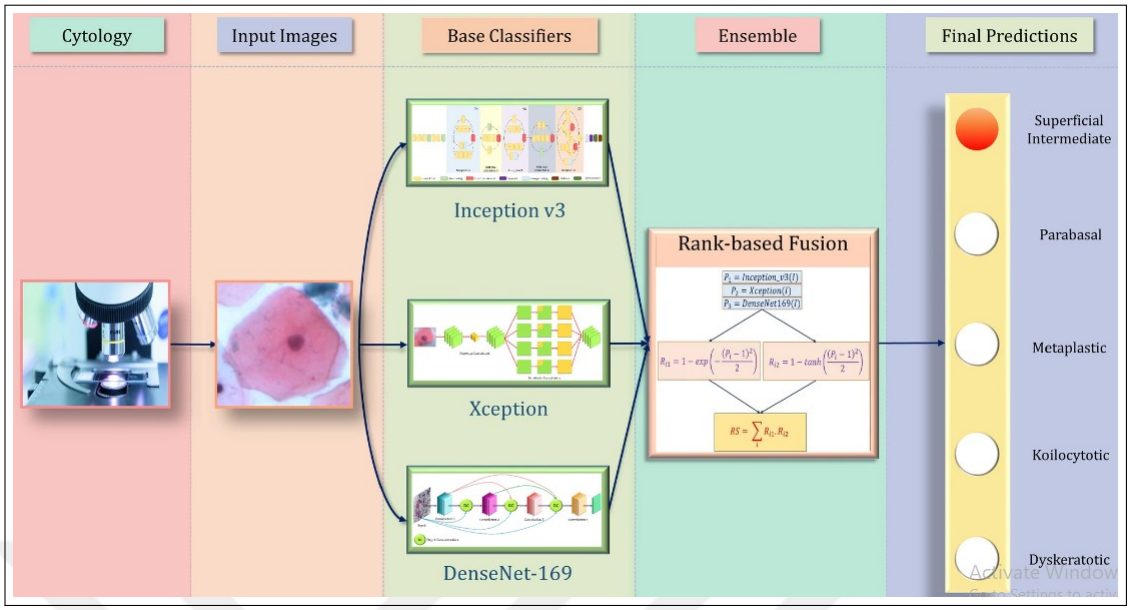
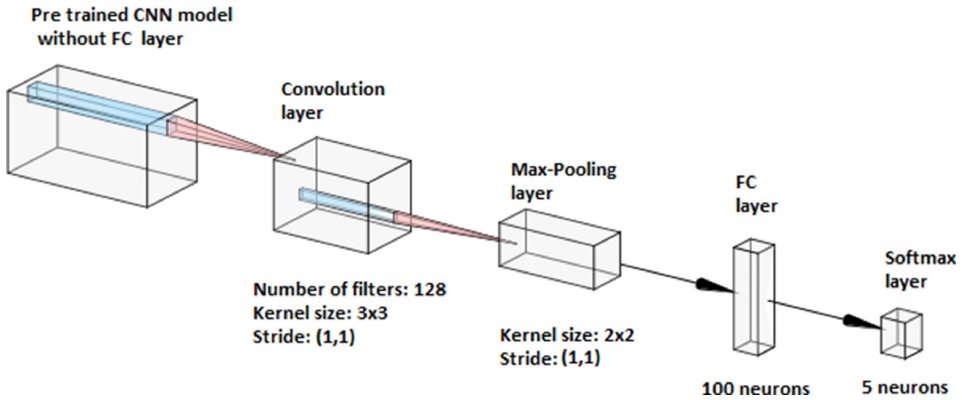


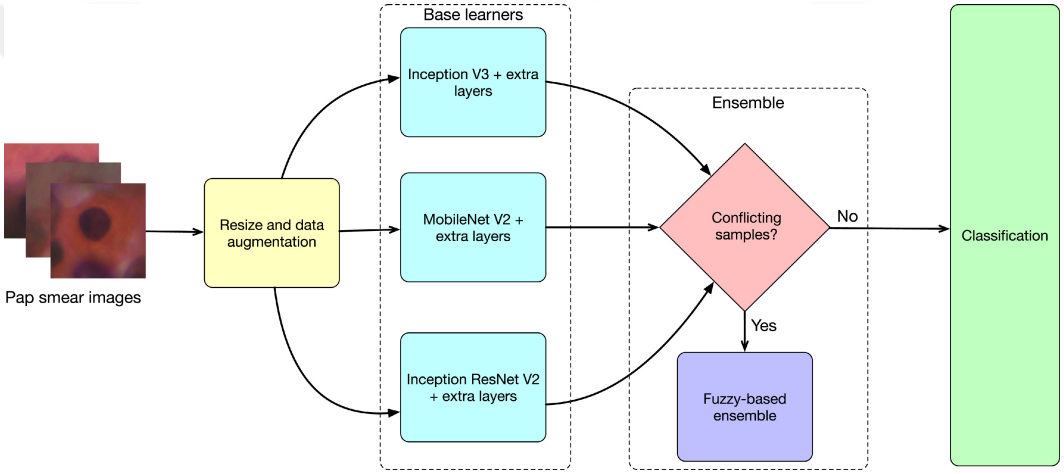

Figure 2.5. The suggested fuzzy rank-based ensemble of CNN models' overall structure, which is utilized to classify cervical cytology [22].

classifiers. The final predictions on the test samples are made using the suggested ensemble approach while considering the accuracy of the core classifiers' predictions. The Mendeley Liquid Based Cytology (LBC) dataset and the SIPaKMeD Pap Smear dataset have been utilized as benchmark datasets to evaluate the suggested technique. On the SIPaKMeD Pap Smear dataset, the proposed framework performs with 98.55% classification accuracy and 98.52% sensitivity in its 2-class configuration and 95.43% accuracy and 98.52% sensitivity in its 5-class configuration. The accuracy and sensitivity of the Mendeley LBC dataset are both 99.23%.

In order to identify cervical cancer in pap smear samples, [23] suggests a fuzzy distance-based ensemble technique built on deep learning models. InceptionV3, MobileNetV2, and InceptionResNetV2 are three transfer learning models used for this purpose, together with extra layers for learning data-specific characteristics. Figure 2.6a demonstrates the extra layers, which include convolution, max-pooling, and fully connected layers proposed in [23]. The outputs of these models are proposed to be aggregated using a novel ensemble approach based on the minimizing of error values between the observed and the ground truth. First, three distance measurements—Cosine, Manhattan (City-Block), and Euclidean—from each class's matching best-case outcome are collected for samples with multiple forecasts. To get the final predictions, these distance measurements are then defuzzified using the product rule.




(a) The deep model's overall pipeline. No padding has been implemented for the convolutional layers, which all utilize the Rectified Linear Unit (ReLU) activation function. The number of output neurons corresponds to the number of classes.




(b) The proposed cervical cancer detection pipeline in its entirety

Figure 2.6. The proposed deep model and cervical cancer detection pipelines [23].

Figure 2.6b depicts the whole pipeline of the technique suggested in [23]. When MobileNet V2, Inception V3, and Inception ResNet V2 were run independently, they achieved 93.92%, 95.30%, and 96.44%, respectively. When the suggested ensemble approach is used, the performance surpasses that of the individual models and reaches 96.96%.

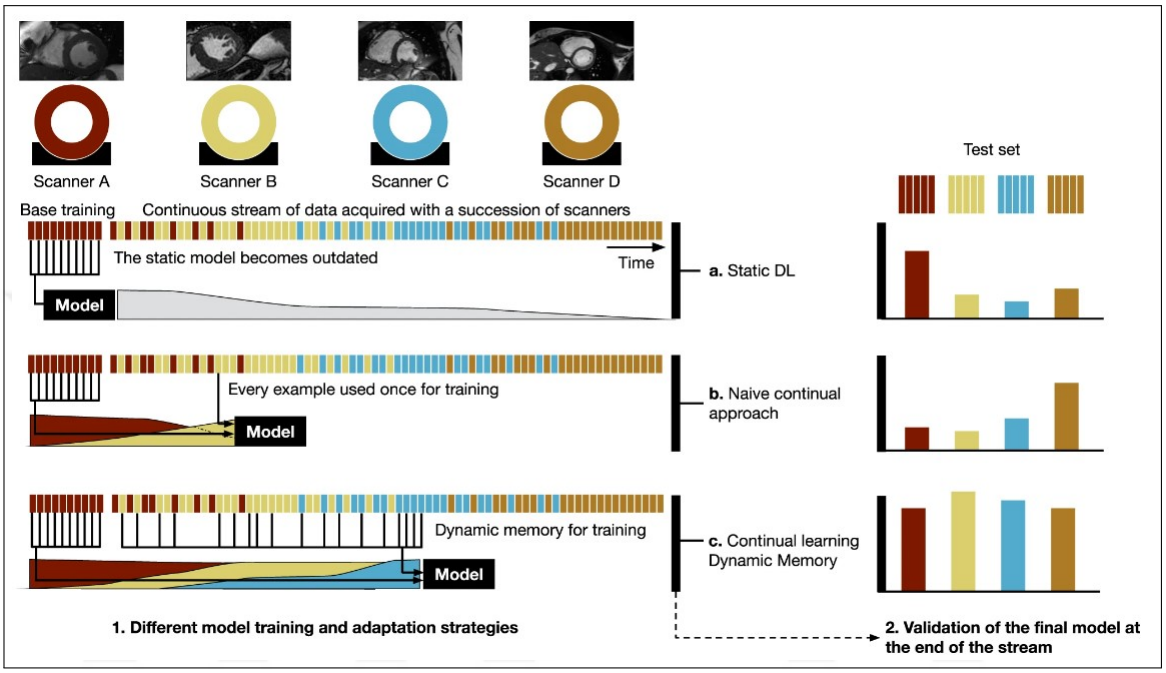



Figure 2.7. Various model training and adaptation strategies [24]

The variety of scanners, ongoing developments in diagnostic techniques, and changing imaging protocols all limit the usefulness of machine learning in clinical settings since they degrade prediction accuracy on new data or render models obsolete. To deal with such domain shifts that occur at unknown time points, [24] proposes a continuous learning approach as shown in Figure 2.7. Models are adjusted to new variances in a continuous data stream while preventing catastrophic forgetting. With the help of dynamic memory, models may be expanded to new domains while practiced on a subset of various training data to reduce forgetting. The approach manages memory use by identifying pseudo-domains, which stand in for various style clusters within the data stream. The technique consistently outperforms competing techniques in lung nodule detection in computed tomography and cardiac segmentation in magnetic resonance imaging.

# 3. THEORETICAL BACKGROUND

This section introduces the cervical screening concept, learning algorithms, and advanced learning technologies used in this thesis.

## 3.1. CERVICAL SCREENING

There are several medical imaging techniques for analyzing and diagnosing various diseases. One of the most widely used screening test for analyzing the cervical cells in order to identify the precancerous lesions is pap smear test. The application of pap smear test includes collecting samples from patient's uterine cervix and analyzing the cervical cells of sample under a microscope. In order to automate this analyzing procedure, the need to digitize the samples has arisen. Recently, several companies manufactured automated scanners which can digitize the samples on several magnifications (10x, 20x, 40x etc).

## 3.2. IMAGE AUGMENTATION

To operate properly, deep networks require a substantial amount of training data. To create a robust image classifier with limited training data, image augmentation is usually necessary [25]. Image augmentation is the artificial creation of training pictures by the use of different processing methods or a mixture of them, such as shifts, random rotation, flips, and shear [26]. This section describes the image augmentation methods used in this study.

### 3.2.1. Affine Transformations

The term "affine transformation" refers to a mathematical operation that translates an affine space onto itself while maintaining the length ratios of parallel line segments as well as the dimensions of any affine subspaces (transferring points, lines, planes, etc. to one another). As a result, after an affine transformation, parallel sets of affine subspaces remain parallel. While ratios of distances between points on a straight line are preserved by an affine translation, lengths between points or angles between lines are not always preserved. In this study,

rotation, scaling, translation, shearing, and flipping (vertical and horizontal) operations are employed.

### 3.2.2. Contrast Limited Adaptive Histogram Equalization

An approach to computer-aided image processing known as Adaptive Histogram Equalization (AHE) is used to enhance contrast in images. The adaptive method is different from conventional histogram equalization in that it computes multiple histograms, each one corresponding to a different area of the image, and then uses them to redistribute the image's lightness values. Therefore, it is appropriate for enhancing the definition of edges in each area of an image as well as the local contrast.

But in relatively homogeneous areas of an image, AHE has a propensity to overamplify noise. By restricting the amplification, the adaptive histogram equalization variant known as Contrast Limited Adaptive Histogram Equalization (CLAHE) avoids this. In this study, CLAHE, all channel CLAHE and gamma contrast are performed.

### 3.2.3. Edge Detection

The input images are converted into edge images using edge detection, which marks non-edge regions as black and edge regions as white.

### 3.2.4. Sharpening

The input images are sharpened and the result overlays with the original image.

### 3.2.5. Canny Filter

The most well-known edge detector and one that models edge detection as a problem of numerical optimization is the Canny edge method [27]. The Canny edge detector employs a filter based on Gaussian's first derivative to reduce visual noise. Since an edge in an image

might point in a number of different directions, the conventional Canny method employs a number of operators to find edges in the noise-filtered picture in four different directions: vertically, horizontally, and in two different diagonal directions. Edge detection operators provide an intensity gradient value due to their use. The likelihood of edges is higher for pixels with big gradient magnitudes than for pixels with lower gradient magnitudes. The Canny method employs high and low thresholds because it is typically not possible to set a threshold of intensity gradient magnitudes to detect edges. The Canny method first uses a high threshold to identify the edges that are most likely to exist. After that, further edges that are probable can be found by tracing predicted directions through the picture while using the low threshold.

### 3.2.6. Channelwise Transformations

Random value between 10 and 100 is added to one of the channels (red, green, blue) of the input image.

### 3.2.7. Applying Noise

A randomly chosen noise among blurring, gaussian noise, and laplace noise is applied to the input image.

### 3.2.8. Photometric Transformations

All the color channels are mixed, images are converted to grayscale, hue and saturation values are changed, more hue and saturation is added, and images are quantized up to 16 different colors.

### 3.2.9. Contrast Adaptation

Contrast adaptation is accomplished by adjusting an image's contrast and brightness.

## 3.3. LEARNING ALGORITHMS

This section describes the learning algorithms employed by the proposed method.The following sections describe the convolution and pooling operations. The classical Artificial Neural Networks and Convolutional Neural Networks, which employ convolution operations, are then discussed.

### 3.3.1. Convolution

Convolution transforms the image by applying a kernel to each pixel and its nearby pixels over the whole image. The convolution process's transformation impact is determined by the size and values of the kernel, which is a matrix of values.

The convolution process consists of the following steps. First, it multiplies each kernel value with the corresponding pixel it is over by placing the kernel matrix over each image pixel, making sure that the entire kernel is contained within the picture. The new value of the center pixel is then obtained by adding the resultant multiplied values. The entire image is subjected to this procedure repeatedly.

The key concept in CNN is convolution. Convolutional layer is one of the layers making up A CNN architecture. Using a kernel matrix that it calibrates during training, a CNN performs convolution to its inputs at the convolution layer. As a result, CNN excels at image feature matching and object classification. A collection of learnable kernels make up the convolution layer parameters. Each kernel consists of a tiny matrix that spans the whole depth of the input volume. Each kernel is convolved over the width and height of the input frame during the forward pass, and dot products are calculated between the source and kernel's pixel values at the appropriate places.

### 3.3.2. Pooling

In CNNs, pooling is a critical step that lowers the dimensionality of the feature maps. It reduces the feature map's dimensionality by combining a group of values into a smaller group

of values. Preserving important information and removing unnecessary information turns the joint feature representation into valuable information. By removing certain connections between convolutional layers, pooling operators offer spatial transformation invariance and lower the computational cost for higher layers.

Pooling layers perform downsampling on the features from the previous layer, resulting in new features with a concise resolution. Pooling layers have two primary functions: First, they lower the computational cost by reducing the number of parameters or weights, and second, they prevent overfitting. An ideal pooling method should extract only useful information while discarding irrelevant details.

Rather than being learned, the pooling operation is specified. The following are two common functions used in the pooling operation:

- **Average Pooling:** Determine each patch's average value on the features. Figure 3.1 presents average pooling example with a 2x2 kernel size.
- **Max Pooling:** Determine the highest value for each features patch. Figure 3.2 presents max pooling example with a 2x2 kernel size.

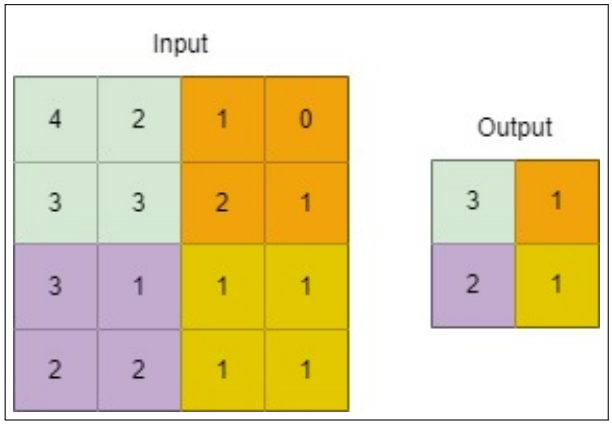



Figure 3.1. Average Pooling

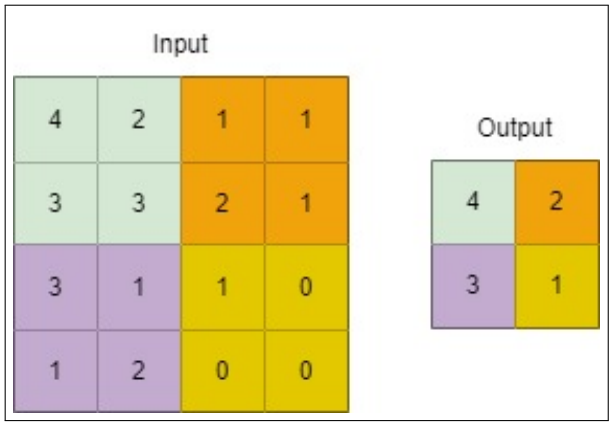



Figure 3.2. Max Pooling

### 3.3.3. Artificial Neural Networks

Biological neurons served as the inspiration for the Artificial Neural Network (ANN), a supervised machine learning method commonly utilized for classification and regression issues. ANNs may be thought of as a network made up of a number of nodes connected by connections.

The Feed Forward Neural Networks are the most basic type of ANN, where each layer's neurons are only coupled to those in the following layer. The input layer, the hidden layer, and the output layer are the three layers that make up a typical feed-forward ANN. Data input is handled by the input layer, while the output layer generates the required output. The hidden layer functions as the transformation layer by converting the input into the desired output format. The data and the issue for which the network is required determine how many hidden layers should be there.

Each node in the layers is referred to as a neuron, and it holds an activation function as well as a weight. The output produced from the weighted inputs is sent to the following node in accordance with the activation function's decision. Each layer's number of neurons is determined by the task and the data.

The weights are modified during training using an update algorithm depending on the output layer error. The training data's labels and the network's anticipated output are used to determine the error. The purpose of this approach is to decrease error so that the network

may create predictions that are as near to the real outputs as feasible.

### 3.3.4. Convolutional Neural Networks

CNN is a deep learning method that interprets images using a customized neural network with multiple layers. CNNs outperform conventional neural networks for image, audio/video, and speech signal inputs due to their superior performance. The main layers used in filtering operations are convolution layers. The convolutional layer is the bottom layer of a convolutional network. The initial layers focus on fundamental elements like colors and borders. More convolutional layers or pooling layers may follow convolutional layers, but the fully-connected layer is the last. With each additional layer, the CNN grows more intricate and can recognize more parts of the image. The CNN detects the item's most relevant elements or features as the visual input moves through the layers, eventually identifying the target object.

## 3.4. ADVANCED LEARNING TECHNOLOGIES

This section describes the CNN architectures and continual learning strategies used in this study.

### 3.4.1. CNN Architectures

This section describes the CNN architectures used in this study. The following sections go into great detail about the InceptionV3 and ResNet50 architectures.

#### *3.4.1.1. InceptionV3*

InceptionV3 architecture consists of a stem block, three types of inception blocks, two types of reduction blocks, an auxiliary classifier and a main classifier. Stem block is the first seven layer of the whole network. It starts with three consecutive 3*3 convolutional layers, followed by a 3*3 max pooling layer, a 1*1 convolutional layer, a 3*3 convolutional layer and one more 3*3 max pooling layer. Figure 3.3 presents the architecture of inceptionV3's stem block.

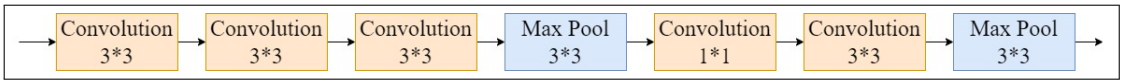



Figure 3.3. Stem Block

The inception blocks are the most distinctive structures of inceptionV3 architecture. There are three kinds of inception blocks in inceptionV3 architecture. All of the inception blocks are consist of four parallel branches.

The first branch of the first kind of inception block is composed of a 1*1 convolutional layer followed by two consecutive 3*3 convolutional layers. The second one has a 1*1 convolutional layer and a 3*3 convolutional layer. The third one starts with a 3*3 average pooling layer followed by a 1*1 convolutional layer. The final branch has only a 1*1 convolutional layer. The result of each branch is concatenated at the end of the block and passed to the next block of the architecture. Figure 3.4 presents the structure of the first type inception block.

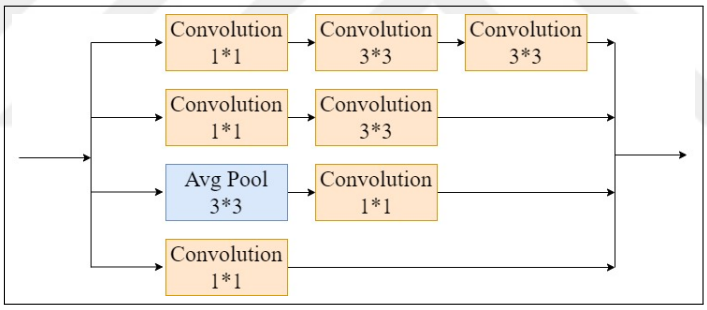



Figure 3.4. Inception Block (Type 1)

The first branch of the second kind of inception block is composed of five consecutive convolutional layers of size 1*1, 7*1, 1*7, 7*1, and 1*7. The second one has a 1*1 convolutional layer followed by a 1*7 convolutional layer and 7*1 convolutional layer consecutively. The third one starts with a 3*3 average pooling layer followed by a 1*1 convolutional layer. The final branch has only a 1*1 convolutional layer. The result of each branch is concatenated at the end of the block and passed to the next block of the architecture. Figure 3.5 presents the structure of the second type inception block.

The first branch of the third type of inception block is composed of a 1*1 convolutional layer followed by a 3*3 convolutional layer and two parallel convolutional layers which are 1*3 and 3*1 convolutional layers. The second branch has a 1*1 convolutional layer and two parallel

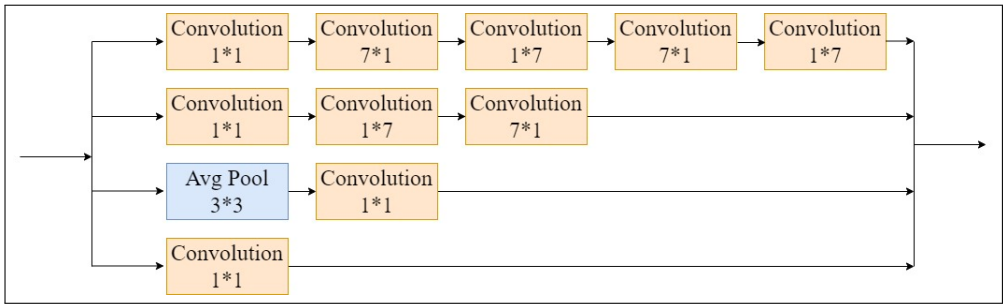



Figure 3.5. Inception Block (Type 2)

convolutional layers as in the first branch. The third branch starts with a 3*3 max pooling layer followed by a 1*1 convolutional layer. The final branch has only a 1*1 convolutional layer. The result of each branch is concatenated at the end of the block and passed to the next block of the architecture. Figure 3.6 presents the structure of the third type inception block.

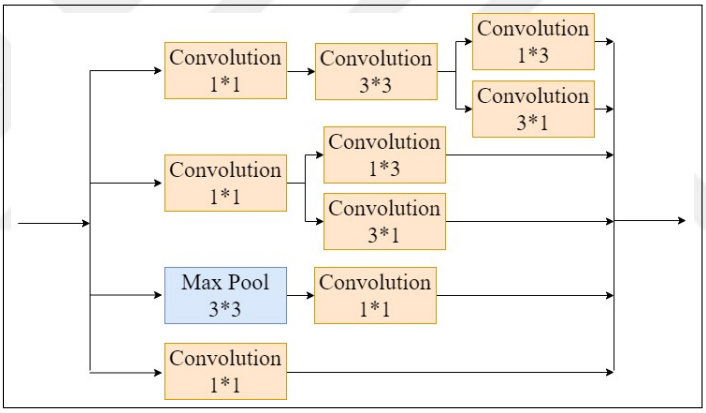



Figure 3.6. Inception Block (Type 3)

The aim of the reduction blocks is to reduce grid size of the feature maps. There are two types of reduction blocks in inceptionV3 architecture. Each of which consists of three branches.

The first branch of the first kind of reduction block is composed of a 1*1 convolutional layer followed by two consecutive 3*3 convolutional layers. The second one has only a 3*3 convolutional layer. The final branch has only a 3*3 max pooling layer. The result of each branch is concatenated at the end of the block and passed to the next block of the architecture. Figure 3.7 presents the structure of the first type reduction block.

The first branch of the second kind of reduction block is composed of four consecutive convolutional layers of size 1*1, 1*7, 7*1, and 3*3. The second one has two consecutive 1*1 convolutional layers. The final branch has only a 3*3 max pooling layer. The result of each

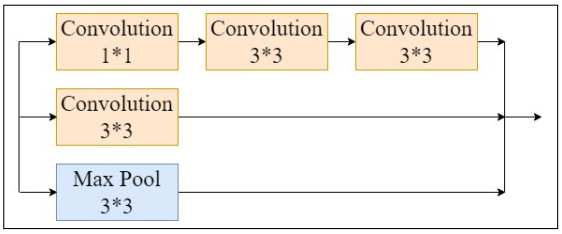



Figure 3.7. Reduction Block (Type 1)

branch is concatenated at the end of the block and passed to the next block of the architecture. Figure 3.8 presents the structure of the second type reduction block.

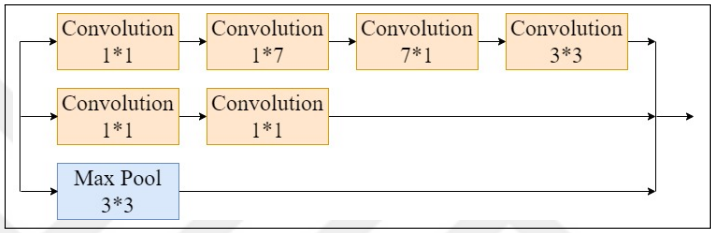



Figure 3.8. Reduction Block (Type 2)

The auxiliary classifier is part of the inceptionV3 architecture but it is not used in transfer learning process. It consists of a 5*5 average pooling layer followed by a 1*1 convolutional layer and two fully connected layers of size 768 and 1000 consecutively. Figure 3.9 presents the structure of the auxiliary classifier.

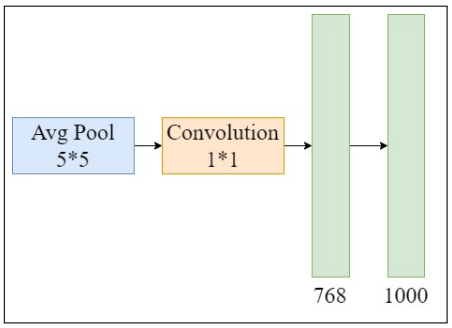



Figure 3.9. Auxiliary Classifier

In overall, the input enters inceptionV3 network from stem block. Three inception type 1 blocks follows the stem block. There is a reduction type 1 block after the inception type 1 blocks. Four inception type 2 blocks, a reduction type 2 block, and two inception type 3 blocks complete the body part of the network used in transfer learning. There are also a global average pooling layer and two consecutive fully connected layers of size 2048 and

1000 consecutively as classifier in the original inceptionV3 network. Figure 3.10 presents the overall architecture of inceptionV3.

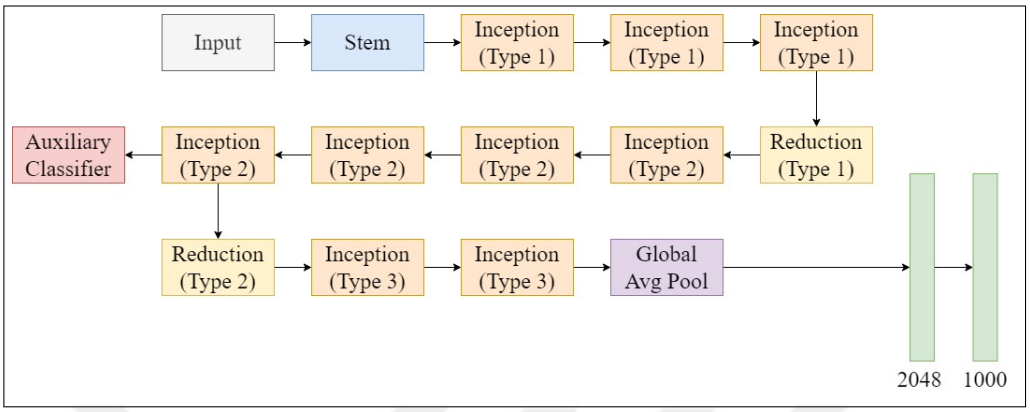



Figure 3.10. InceptionV3 Overall Architecture

### *3.4.1.2. ResNet50*

ResNet50 architecture has 50 layers in total. Almost all layers are part of the convolutional block or identity block. The convolutional block has two parallel branches. One of them has a 3*3 convolutional layer between two 1*1 convolutional layer consecutively while the other branch has only a 1*1 convolutional layer. The outputs of these two branches are summed at the end of the block. Figure 3.11 presents the structure of the convolutional block.

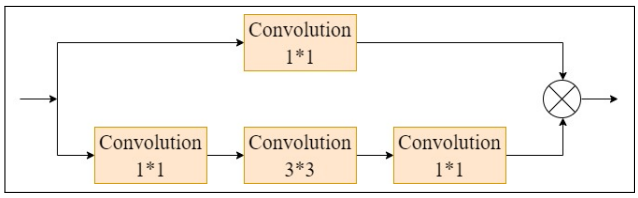



Figure 3.11. Convolutional Block

The identity block has also two parallel branches. One of them has a 3*3 convolutional layer between two 1*1 convolutional layer consecutively as in the convolutional block. The other block on the other hand, has nothing but the input of the block is added to the output of the first branch at the end of the block. Figure 3.12 presents the structure of the convolutional block.

In overall, the input enters ResNet50 network from a 7*7 convolutional layer. 3*3 max pooling layer follows the convolutional layer before the convolutional and identity block

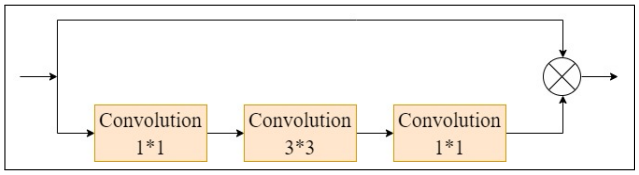



Figure 3.12. Identity Block

series. After the max pooling layer, there are four convolutional blocks each of them followed by two, three, five, and two identity blocks consecutively. There are also a global average pooling layer and a fully connected layer of size 1000 but they are not used in the transfer learning phase. Figure 3.13 presents the overall architecture of ResNet50.

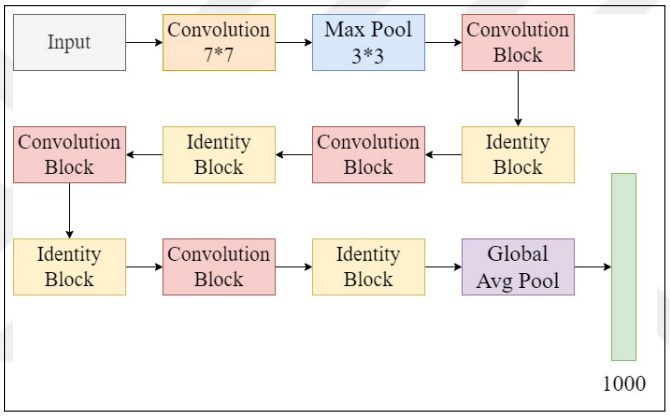



Figure 3.13. ResNet50 Overall Architecture

### 3.4.2. Continual Learning

This section describes the strategies for continual learning used in this study. The sections that follow present naive and rehearsal strategies.

#### *3.4.2.1. Naive Strategy*

The simplest and least effective Continual Learning strategy is the naive strategy. Naive fine-tunes a single model incrementally without employing any method to contrast the catastrophic forgetting of prior knowledge.

Naive strategy is also known as sequential learning, described as human-like learning as opposed to typical concurrent learning in neural networks in [14]. Performance on previously learned things might be severely hampered by training on a fresh set of objects during the

sequential learning.

In sequential training, the neural networks are trained on a group of objects first and then on a different set in order to evaluate how well the previously learned items were retained while learning the new ones. Naive is simple to set up, and its results are frequently used to demonstrate the worst performing baseline.

#### *3.4.2.2. Rehearsal Strategy*

When a new object is presented, a portion of the previously taught material is reviewed during rehearsal. It is conceivable that this procedure, which makes use of the resilience of training items in sets, might "enhance" formerly learned things or "defend" them against interruption by new items. In [28], the effectiveness of a straightforward rehearsal mechanism was tested for minimizing the catastrophic forgetting effect. At each experience in the rehearsal strategy, the model is trained using data from all previous and current experiences.

#### *3.4.2.3. Elastic Weight Consolidation*

EWC is a neural network approach that functions by restricting critical parameters to remain close to their previous values. EWC maintains the performance of the old task while learning a new task by restricting the parameters to stay in a zone of low error for the old task, as shown schematically in Figure 3.14 [29].

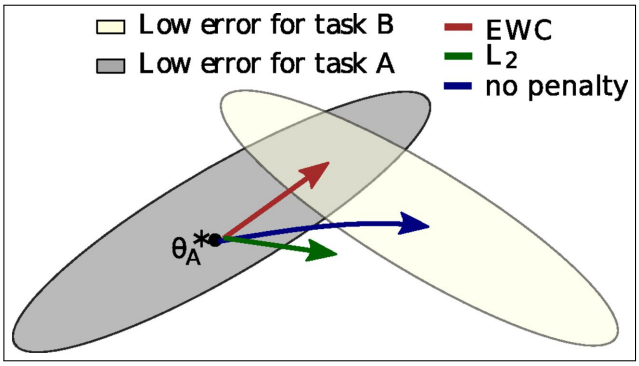



Figure 3.14. When practicing task B, EWC makes sure task A is retained. EWC handles task B without a substantial loss on task A. In a schematic representation of parameter space, training trajectories are shown, and parameter regions that result in successful completion of tasks A and B are highlighted. [29].

EWC reduces learning for task-relevant weights coding for previously learned knowledge by imposing a quadratic penalty on the difference between parameters for the old and new tasks.

The posterior distribution $p(\theta \mid D)$ models the importance of the parameter $\theta$ with regard to the training data $D$ for a task. The posterior probability given by the Bayes' rule, assuming a situation with two independent tasks, $A$ with $D_A$ and $B$ with $D_B$, is:

$$logp(\theta \mid D) = logp(D_B \mid \theta) + logp(\theta \mid D_A) - logp(D_B) \tag{3.1}$$

where the posterior probability $logp(\theta \mid D_A)$ embeds all the data from the prior task.

The loss function used in EWC is:

$$L(\theta) = L_B(\theta) + \sum_i \frac{\lambda}{2} F_i(\theta_i - \theta^*_{A,i})^2 \tag{3.2}$$

where $L_B(\theta)$ is the calculated loss for just task B, $i$ denotes each trainable parameter in the network, and $\lambda$ determines importance of the former task comparing the new one. EWC will attempt to retain the network parameters near to the parameters that were learned for tasks A and B while switching to task C.

# 4. METHODOLOGY

In this study, the InceptionV3 and ResNet50 architectures described in the previous chapter are used for the cervical cell binary classification task. Continuous learning strategies described in the previous chapter are used in the training process to address catastrophic forgetting and ways to avoid it.

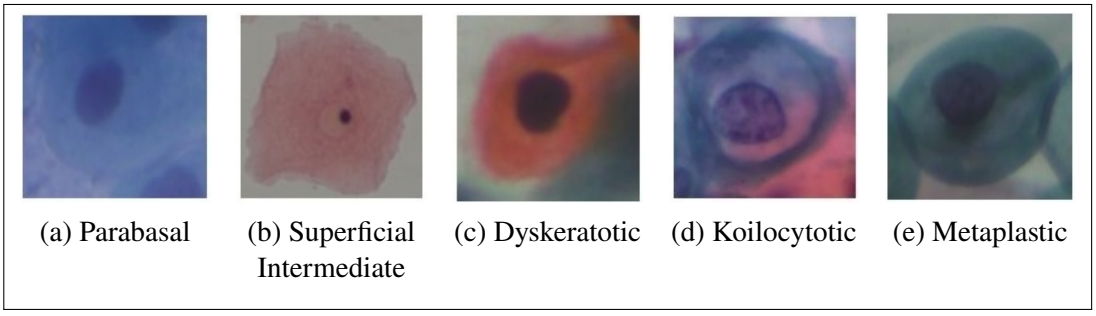
The following sections describe the data and augmentation methods used in this study.

## 4.1. DATASETS

In order to apply and test continual learning strategies, more than one dataset is needed. Therefore, three different datasets, namely Sipakmed, Herlev, and CRIC datasets, are used in this study.

### 4.1.1. Sipakmed Dataset

4049 single-cell pictures and 966 pap smear slide photos are included in this collection. The cells are divided into five groups based on morphology and cell structure. They are parabasal, superficial-intermediate, dyskeratotic, koilocytotic, and metaplastic. Normal cells are the first two classes. The following two groups of cells are considered abnormal, as morphological alterations in the nucleus begin to occur. Metaplastic, the last classification, is the stage of the benign cell, where precancerous and cancerous abnormalities of the cervix first appear [30]. Figure 4.1 presents examples of each category in Sipakmed dataset [30].



(a) Parabasal (b) Superficial Intermediate (c) Dyskeratotic (d) Koilocytotic (e) Metaplastic

Figure 4.1. Sample images for each 5 category of Sipakmed database

Table 4.1. Sipakmed Database

| Category | Cell Type | Number of Cells | Subtotals |
|---|---|---|---|
| Normal | Parabasal | 787 | 1618 |
| | Superficial-Intermediate | 831 | |
| Abnormal | Dyskeratotic | 813 | 1638 |
| | Koilocytotic | 825 | |
| Benign | Metaplastic | 793 | 793 |
| Total | | 4049 | |

### 4.1.2. Herlev Dataset

The personnel at Herlev University Hospital in Denmark created and examined the pap smear benchmark database. They created a dataset including 500 single cell images (150 normal, 350 abnormal) in 2003. Then, they made another dataset in 2005 containing 917 single cell images (242 benign, 675 malignant). The database is divided into seven distinct categories: severe dysplasia, mild squamous non-keratinizing dysplasia, moderate dysplasia, carcinoma in situ, columnar epithelia, superficial squamous epithelia, and intermediate squamous epithelia. The first four of those seven classes belong to diseased cells, while the remaining classes correspond to healthy cells [31]. Figure 4.2 presents examples of each category in Herlev dataset [32].

Table 4.2. Herlev Database

| Category | Cell Type | Number of Cells | Subtotals |
|---|---|---|---|
| Normal | Superficial squamous epithelial | 74 | 242 |
| | Intermediate squamous epithelial | 70 | |
| | Columnar epithelial | 98 | |
| Abnormal | Mild squamous non-keratinizing dysplasia | 182 | 675 |
| | Moderate squamous non-keratinizing dysplasia | 146 | |
| | Severe squamous non-keratinizing dysplasia | 197 | |
| | Squamous cell carcinoma in situ intermediate | 150 | |
| Total | | 917 | |

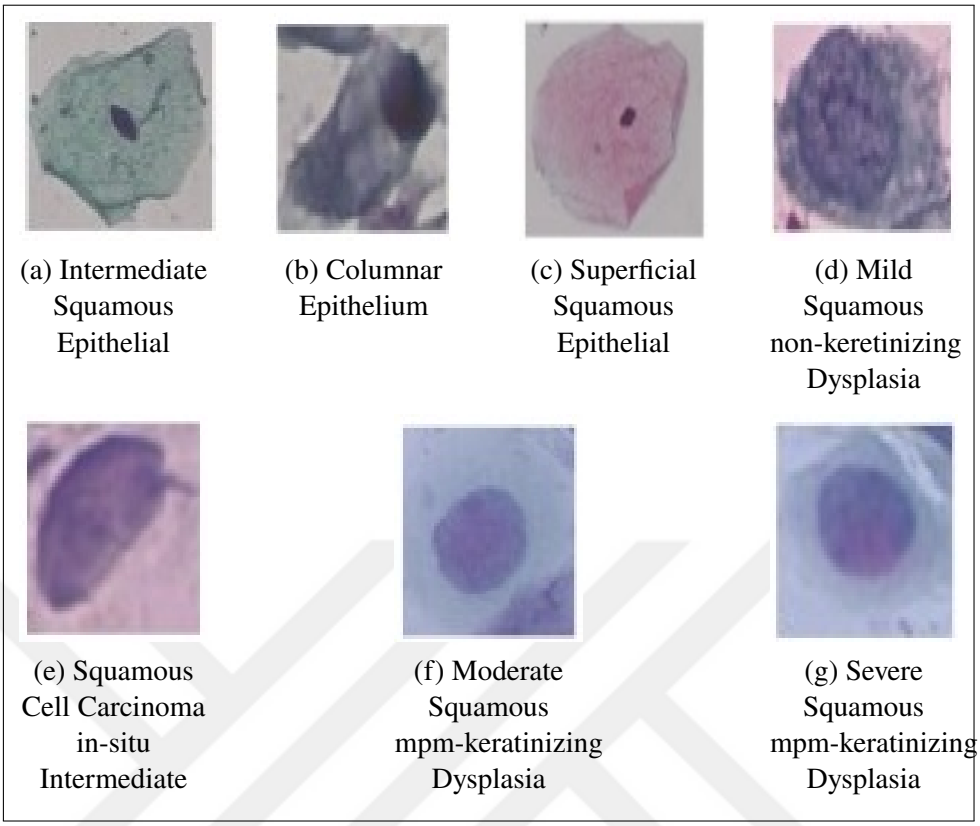



Figure 4.2. Sample images for each 7 category of Herlev database

### 4.1.3. CRIC Dataset

There are 11,534 categorized cells and 400 Pap smear images in the CRIC Cervix collection. The six classes of cells in the CRIC Cervix collection are as follows: High-grade Squamous Intraepithelial Lesion (HSIL), Low-grade Squamous Intraepithelial Lesion (LSIL), Atypical Squamous Cells, cannot exclude a High-grade lesion (ASC-H), Atypical Squamous Cells of Unclear Significance (ASC-US), Squamous Cell Carcinoma (SCC), and Negative for Intraepithelial Lesion or Malignancy (NILM). In terms of binary categorization, the only cells that fall into the normal category are NILM cells, while the lesions that fall into the abnormal cell category are ASC-US, LSIL, ASC-H, HSIL, and SCC lesions [33]. Figure 4.3 presents examples of each category in CRIC dataset [34].

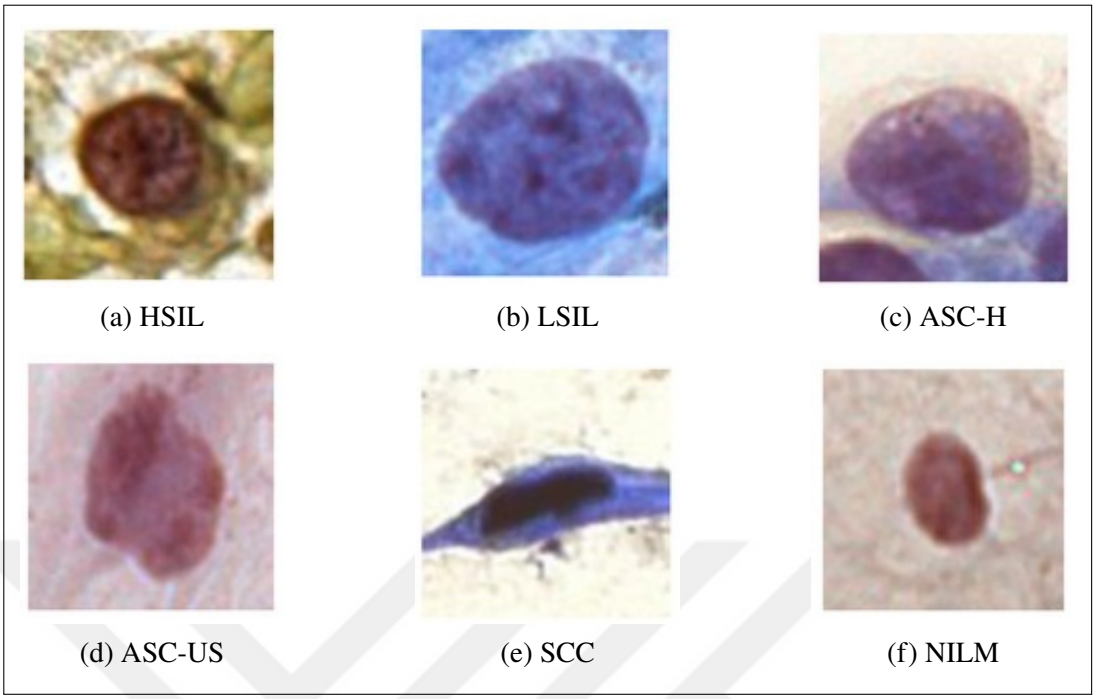



Figure 4.3. Sample images for each 6 category of CRIC database

Table 4.3. CRIC Database

| Category | Cell Type | Number of Cells | Subtotals |
|---|---|---|---|
| | HSIL | 1703 | |
| | LSIL | 1360 | |
| Abnormal | ASC-H | 925 | 4755 |
| | ASC-US | 606 | |
| | SCC | 161 | |
| Normal | NILM | 6779 | 6779 |
| Total | | 11534 | |

## 4.2. DATASET PREPARATION

Sipakmed, Herlev, and CRIC datasets are went through some processing in order to be prepared for the training phase. These processes can be examined in three parts.

First, the single cell images of Sipakmed, Herlev, and CRIC datasets are all in various shapes. In order to standardize and fit their shapes to the inceptionV3 and ResNet50 architectures' input shapes, rescaling was applied to all the data. All images were reshaped as 299x299 pixels for inceptionV3 and 224x224 for ResNet50 architecture. Then, the Herlev and Sipakmed datasets were split into training, validation, and test datasets with 60%, 20%, and 20%, respectively. Finally, the training data went through some augmentation processes described in the previous chapter to increase the amount required for the training process and balance the data between categories. For each particular image, random augmentation methods were selected and applied.

Table 4.4. The Sipakmed dataset's experimental data settings.

| Dataset | Number of Images | | |
|---|---|---|---|
| | Abnormal | Normal | Total |
| Training | 6874 | 6790 | 13664 |
| Validation | 328 | 324 | 652 |
| Test | 328 | 324 | 652 |

Table 4.5. The Herlev dataset's experimental data settings.

| Dataset | Number of Images | | |
|---|---|---|---|
| | Abnormal | Normal | Total |
| Training | 6075 | 2160 | 8235 |
| Validation | 135 | 49 | 184 |
| Test | 135 | 49 | 184 |

The splitting and augmentation processes were applied to Herlev and Sipakmed datasets only. Since the CRIC dataset was already split into training, validation, and test sets and augmented, this dataset was used as it is. Table 4.4, Table 4.5, and Table 4.6 show the distribution of the Sipakmed, Herlev, and CRIC data used in the experiments respectively.

Table 4.6. The CRIC dataset's experimental data settings.

| Dataset | Number of Images | | |
|---|---|---|---|
| | Abnormal | Normal | Total |
| Training | 2758 | 2756 | 5514 |
| Validation | 692 | 689 | 1381 |
| Test | 477 | 173 | 650 |

## 4.3. MODELS

Two alternative pre-trained models and a custom model were built to train the smear images for the binary classification problem.

### 4.3.1. Pre-trained Models

The pre-trained models have two primary parts: The body and the head parts. The body component was utilized to extract features, while the head part was used for classification. The architectures "ResNet50" and "InceptionV3" were utilized as body components of models with the same structure as stated in the previous chapter. The input layer shapes for "ResNet50" are 224x224 and 299x299 for "InceptionV3."

The identical head was built and utilized for both networks. The head section is made up of dropout and fully connected layers. To avoid overfitting, a dropout layer was added to both networks. Finally, a fully connected layer with two outputs was implemented as a classifier using a softmax activation function to provide predictions. The structure of the proposed pre-trained models for applying continual learning strategies is shown in Figures 4.4a and 4.4b.

### 4.3.2. Custom CNN

The custom neural network proposed in this thesis consists of two convolutional layers and two fully connected layers. The network starts with a convolutional layer with 32 filters and another convolutional layer with 64 filters. After these two convolutional layers, there is a

dropout before the two fully connected layers. The first of the fully connected layers has 1024 input and 32 output, whereas the last one has 32 input and two output units, which is the number of output classes of the model. There is one more dropout between two fully connected layers to prevent overfitting. Figure 4.4c depicts the structure of the proposed custom model for implementing continuous learning strategies.

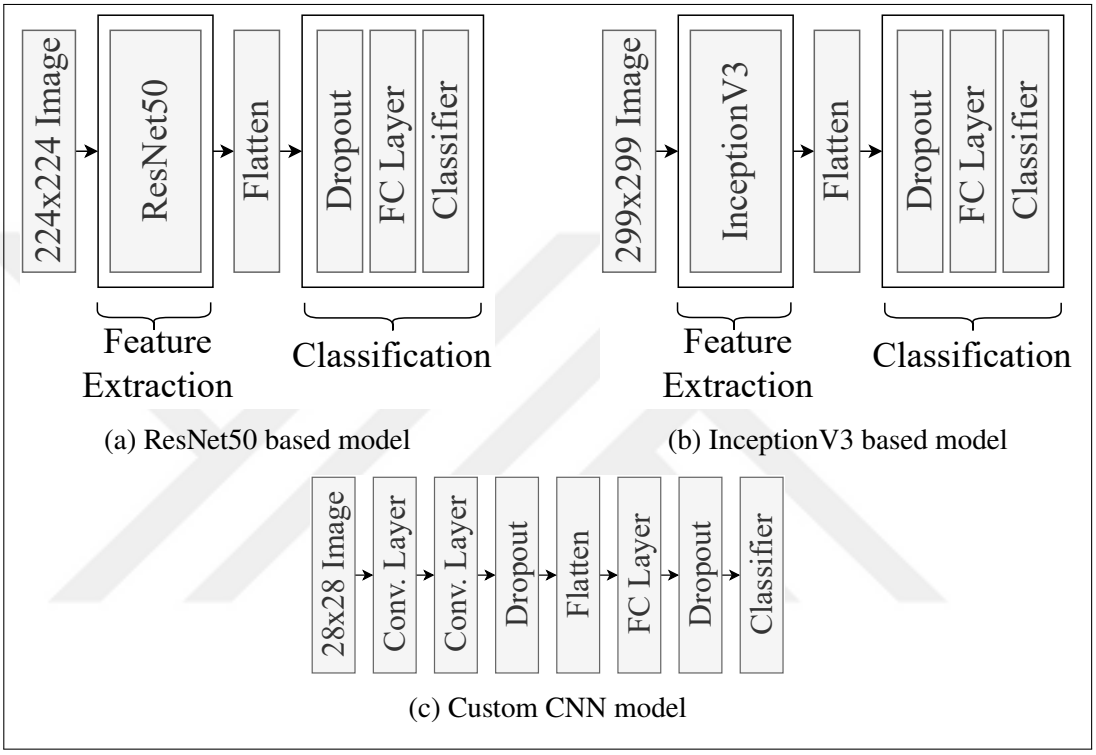



Figure 4.4. The structure of proposed models

## 4.4. CONTINUAL LEARNING STRATEGIES

The constructed models were trained in four different ways using the Continual Learning (CL) techniques mentioned in the preceding chapter.

### 4.4.1. Naive Strategy

First, the models were successively trained over three datasets (Sipakmed, Herlev, and CRIC). In other words, the models were trained on the Sipakmed dataset for a certain number of epochs, then on the Herlev dataset for the same number of epochs, and finally on the CRIC

dataset for the same number of epochs. This is referred to as the naïve strategy.

### 4.4.2. Rehearsal Strategy

The models were trained using cumulatively aggregated datasets for the second part of the experiments. In other words, the models were trained on the Sipakmed dataset for a specific number of epochs, then on the Sipakmed and Herlev datasets combined, and lastly on the Sipakmed, Herlev, and CRIC datasets together. This is referred to as the rehearsal strategy.

### 4.4.3. EWC Strategy

The experiments went on with the same training approach as the naïve technique. Additionally, after training on a particular dataset, the model's parameter values were saved, and the Fisher information matrix was generated. The difference between the new and stored parameter values was then multiplied by the Fisher information matrix and a lambda factor during training on the following dataset. The result was added to the loss produced by the model's standard loss function to compute the new loss, which is used to update the weights. This is known as the Elastic Weight Consolidation technique.

### 4.4.4. Rehearsal EWC Strategy

The final strategy simply combines the rehearsal and the EWC strategies. The models were trained using datasets that had been cumulatively aggregated. The model's parameters were saved and fisher information matrix was generated after each training phase. The loss was calculated according to EWC procedure.

## 4.5. EVALUATION METHOD

Accuracy, specificity, sensitivity, precision, and F1 score are the most commonly used metrics for assessing classification performance. All of these metrics are derived from the confusion matrix.

### 4.5.1. Confusion Matrix

In order to solve classification difficulties, confusion matrices are frequently utilized. Both binary classification and multiclass classification issues can benefit from their use. Table 4.7 provides an illustration of a binary classification confusion matrix.

Table 4.7. Binary classification confusion matrix

| | | Predicted Class | |
|---|---|---|---|
| True | True Positive (TP) | False Negative (FN) |
| Class | False Positive (FP) | True Negative (TN) |

The counts from the expected and actual values are represented by confusion matrices. The True Negative indicator displays the quantity of appropriately categorized negative samples. The quantity of precisely categorized positive instances is shown by True Positive in a similar manner. The number of actual negative cases categorized as positive is referred to as False Positive, while the number of genuine positive examples labeled as negative is referred to as False Negative.

### 4.5.2. Accuracy

Accuracy is the percentage of correct predictions out of all forecasts. Equation (4.1) is used to calculate it.

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN} \tag{4.1}$$

### 4.5.3. Specificity

Specificity is defined as the proportion of negative values among all genuine negative cases. To put it another way, it refers to the percentage of genuine negative situations that are accurately detected. Equation (4.2) is used to calculate it.

$$Specificity = \frac{TN}{TN + FP} \tag{4.2}$$

### 4.5.4. Recall (Sensitivity)

Recall is the percentage of positive values out of all actual positive instances. Equation (4.3) is used to calculate it.

$$Recall = \frac{TP}{TP + FN} \tag{4.3}$$

### 4.5.5. Precision

Precision is the percentage of predicted positive values that are positive. In other words, precision is the proportion of correctly identified positive values. Equation (4.4) is used to calculate it.

$$Precision = \frac{TP}{TP + FP} \tag{4.4}$$

### 4.5.6. F1 Score

The harmonic mean of precision and recall is the F1 score. It emphasizes both metrics. Equation (4.5) is used to calculate it.

$$F1Score = 2 \times \frac{Precision \times Recall}{Precision + Recall} \tag{4.5}$$

### 4.5.7. McNemar's Test

McNemar's test [35] is based on a 2x2 contingency table, as seen in Table 4.8. The null hypothesis asserts that classifiers 1 and 2 will correctly classify the same proportion of samples. $H_0 : B = C$ is the null hypothesis. The test is based on the chi-square statistic, which is calculated as stated in Equation (4.6). The p value is obtained using the $\chi^2$ cumulative distribution function.

$$\chi^2 = \frac{(B - C)^2}{B + C} \tag{4.6}$$

Table 4.8. Contingency table for McNemar's test

| | Classifier 2 Hits | Classifier 2 Misses |
|---|---|---|
| Classifier 1 Hits | A | B |
| Classifier 1 Misses | C | D |

# 5. IMPLEMENTATION

All processes are performed on a computer equipped with a 64-bit Ubuntu operating system, 8 GB of RAM, an Intel i7 processor, and a GeForce GTX 970 graphics card. The framework is built using the Python programming language's PyTorch library. All models are executed in a GPU environment (Cuda environment).

The datasets were divided into training, validation, and test sets using the "Scikit-learn" package. First, the datasets were divided into 80% and 20% train and test sets. The train sets were then divided into 75% train sets and 25% validation sets. As a result, the entire dataset was divided into 60% training, 20% validation, and 20% test sets.

After separating the datasets, the "imgaug" package was used to apply image augmentation algorithms to the training and validation sets. After that, the freshly produced images were saved alongside the originals.

The models were created with the "pytorch" package, with the original configuration of the "ResNet50" and "InceptionV3" architectures serving as the models' bodies. To avoid overfitting, a dropout layer with a probability of 50% was added to both networks. Finally, a fully connected layer with two outputs was implemented as a classifier using a softmax activation function to provide predictions.

Stochastic Gradient Descent (SGD) was used as the optimizer for the training process, with a learning rate of $10^{-3}$ and a momentum of 0.9. The base loss function was "cross-entropy loss." The cross-entropy loss was used in the naive and rehearsal strategies and the first phase of the EWC strategy. In the subsequent phases of the EWC approach, this loss function was changed based on the parameter relevance of networks. The training data were loaded for training with a batch size of 16, and the models were trained for 50 epochs for all phases of all strategies. The best weights were saved after 50 epochs based on validation accuracy.

The evaluation techniques, including accuracy, specificity, sensitivity, precision, and f1 score were built using the "scikit-learn" package. The "matplotlib" package was used to visualize the evaluations as bar charts. Finally, "mlxtend" package was used to apply the McNemar's test to classifiers.

# 6. RESULTS

In this thesis, the proposed models for networks and the effects of various continuous learning strategies are evaluated. The assessment is divided into two parts: network model analysis and continual learning strategy analysis. The Sipakmed, Herlev, and CRIC test sets are used to evaluate each strategy-model combination. The average values of the metrics obtained from each of these three tests are presented in the performance measures. To compare the models, the classification accuracy, specificity, sensitivity, precision, and f1 score values are used. McNemar's test was used to compare the models and strategies pairwisely.

## 6.1. NETWORK MODEL ANALYSIS

The network model analysis includes the results of three network models used in this study: Resnet50, InceptionV3, and 2-layer CNN. Each model has been trained with Naive, Rehearsal, EWC, and Rehearsal-EWC strategies and analyzed among these strategies separately.

### 6.1.1. Naive Strategy

Table 6.1. Comparative performance metrics of Resnet50, InceptionV3 and 2-layer CNN model with Naive strategy

| Model | Test Dataset | Metric | | | | |
|---|---|---|---|---|---|---|
| | | Accuracy | Specificity | Sensitivity | Precision | F1 Score |
| Resnet50 | Sipakmed | 0.63 | 0.79 | 0.48 | 0.70 | 0.57 |
| | Herlev | 0.78 | 0.88 | 0.75 | 0.94 | 0.83 |
| | CRIC | 0.97 | 0.93 | 0.98 | 0.98 | 0.98 |
| InceptionV3 | Sipakmed | 0.62 | 0.86 | 0.38 | 0.74 | 0.51 |
| | Herlev | 0.58 | 0.80 | 0.50 | 0.87 | 0.63 |
| | CRIC | 0.84 | 0.79 | 0.86 | 0.92 | 0.89 |
| 2-Layer CNN | Sipakmed | 0.78 | 0.74 | 0.82 | 0.76 | 0.79 |
| | Herlev | 0.77 | 0.49 | 0.87 | 0.82 | 0.84 |
| | CRIC | 0.94 | 0.87 | 0.96 | 0.95 | 0.96 |

The accuracy, specificity, sensitivity, precision, and f1-score measures obtained from

Resnet50, InceptionV3, and 2-layer CNN models using a naive strategy are shown in Table 6.1. 2-layer CNN outperforms the other models in terms of accuracy, sensitivity, precision, and f1 score for the Sipakmed test set, with 0.78, 0.82, 0.76, and 0.79, respectively. In specificity, however, InceptionV3 outperforms other models with a score of 0.86. Resnet50 outperforms all other models for the Herlev test set, with 0.78, 0.88, and 0.94, respectively. The sensitivity and f1-scores of 2-layer CNN are higher than those of the other models, at 0.87 and 0.84, respectively. Resnet50 outperforms all other models in all performance metrics for the CRIC test set, with 0.97 accuracy, 0.93 specificity, 0.98 sensitivity, 0.98 precision, and 0.98 f1 score.

When the models are trained using a naive strategy, the 2-layer CNN has the highest accuracy, sensitivity, and f1 score based on the average of all particular performance metrics from the Sipakmed, Herlev, and CRIC tests. Resnet50, on the other hand, has the highest specificity and precision, as shown in Figure 6.1. McNemar's test finds no statistical difference between Resnet50 and 2-layer CNN ($p=0.18$), but both outperform InceptionV3 ($p<0.05$).

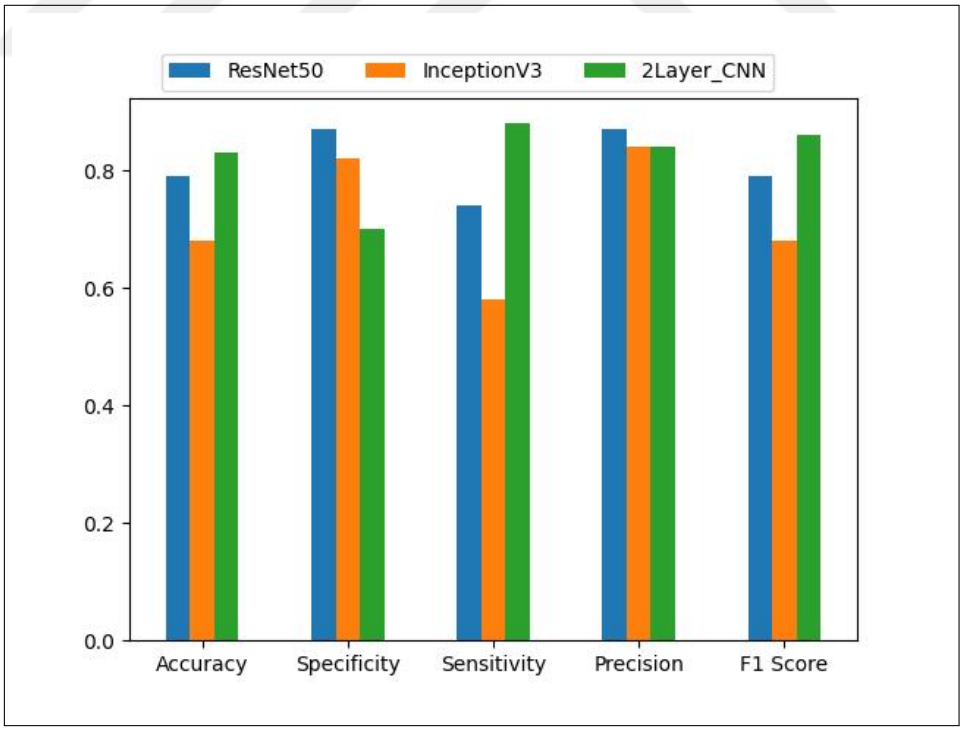



Figure 6.1. The comparison of Resnet50, InceptionV3, and 2-layer CNN models with naive strategy

### 6.1.2. Rehearsal Strategy

Table 6.2. Comparative performance metrics of Resnet50, InceptionV3 and 2-layer CNN model with Rehearsal strategy

| Model | Test Dataset | Metric | | | | |
|---|---|---|---|---|---|---|
| | | Accuracy | Specificity | Sensitivity | Precision | F1 Score |
| Resnet50 | Sipakmed | 0.92 | 0.90 | 0.95 | 0.90 | 0.93 |
| | Herlev | 0.80 | 0.67 | 0.84 | 0.88 | 0.86 |
| | CRIC | 0.89 | 0.84 | 0.91 | 0.94 | 0.92 |
| InceptionV3 | Sipakmed | 0.82 | 0.71 | 0.92 | 0.76 | 0.84 |
| | Herlev | 0.85 | 0.57 | 0.96 | 0.86 | 0.91 |
| | CRIC | 0.81 | 0.65 | 0.87 | 0.87 | 0.87 |
| 2-Layer CNN | Sipakmed | 0.95 | 0.94 | 0.97 | 0.94 | 0.95 |
| | Herlev | 0.86 | 0.59 | 0.96 | 0.87 | 0.91 |
| | CRIC | 0.92 | 0.82 | 0.95 | 0.94 | 0.94 |

The performance measures of the same models with the rehearsal strategy are presented in Table 6.2. With 0.95 accuracy, 0.94 specificity, 0.97 sensitivity, 0.94 precision, and 0.95 f1 score, 2-layer CNN outperforms all other models in all performance metrics for the Sipakmed test set. For the Herlev test set, 2-layer CNN outperforms other models in terms of accuracy with 0.86, and shares the best scores in terms of sensitivity and f1 score with InceptionV3 with 0.96 and 0.91, respectively. Resnet50, on the other hand, outperforms other models in specificity and precision, with 0.67 and 0.88, respectively. For the CRIC test set, 2-layer CNN outperforms other models in terms of accuracy, sensitivity, and f1 score, with 0.92, 0.95, and 0.94, respectively. Resnet50 outperforms other models in specificity with 0.84 and shares the best precision score with 2-layer CNN with 0.94.

The 2-layer CNN has the highest accuracy, sensitivity, precision, and f1 score when trained using the rehearsal strategy, as shown in Figure 6.2. Resnet50 has the highest specificity. McNemar's test shows that 2-layer CNN outperforms the Resnet50 and InceptionV3 models ($p<0.05$), while Resnet50 outperforms the InceptionV3 model ($p<0.05$).

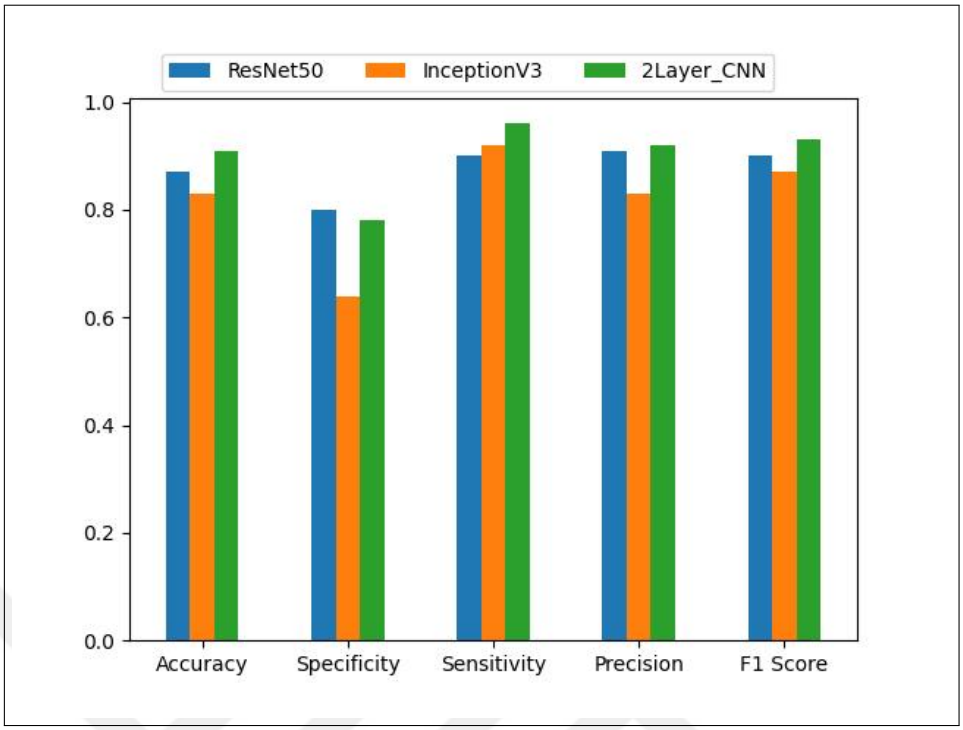



Figure 6.2. The comparison of Resnet50, InceptionV3, and 2-layer CNN models with rehearsal strategy

### 6.1.3. EWC Strategy

Table 6.3 shows the performance metrics for the same models using the EWC strategy. 2-layer CNN outperforms all other models in all performance metrics for the Sipakmed test set, with 0.76 accuracy, 0.85 specificity, 0.68 sensitivity, 0.82 precision, and 0.74 f1 score. Resnet50 outperforms other models in all performance metrics for both the Herlev and CRIC test sets, with 0.85 accuracy, 0.73 specificity, 0.89 sensitivity, 0.90 precision, and 0.90 f1 score for Herlev; 0.96 accuracy, 0.95 specificity, 0.97 sensitivity, 0.98 precision, and 0.97 f1 score for CRIC test set.

When trained using the EWC strategy, the Resnet50 has the highest average score for all performance metrics, as shown in Figure 6.3. According to McNemar's test, there is no statistical difference between Resnet50 and 2-layer CNN ($p=0.19$), but both outperform InceptionV3 ($p<0.05$).

Table 6.3. Comparative performance metrics of Resnet50, InceptionV3 and 2-layer CNN model with EWC strategy

| Model | Test Dataset | Metric | | | | |
|---|---|---|---|---|---|---|
| | | Accuracy | Specificity | Sensitivity | Precision | F1 Score |
| Resnet50 | Sipakmed | 0.72 | 0.78 | 0.66 | 0.75 | 0.70 |
| | Herlev | 0.85 | 0.73 | 0.89 | 0.90 | 0.90 |
| | CRIC | 0.96 | 0.95 | 0.97 | 0.98 | 0.97 |
| InceptionV3 | Sipakmed | 0.58 | 0.71 | 0.45 | 0.61 | 0.52 |
| | Herlev | 0.56 | 0.69 | 0.51 | 0.82 | 0.63 |
| | CRIC | 0.71 | 0.49 | 0.79 | 0.81 | 0.80 |
| 2-Layer CNN | Sipakmed | 0.76 | 0.85 | 0.68 | 0.82 | 0.74 |
| | Herlev | 0.79 | 0.59 | 0.87 | 0.85 | 0.86 |
| | CRIC | 0.90 | 0.86 | 0.91 | 0.95 | 0.93 |

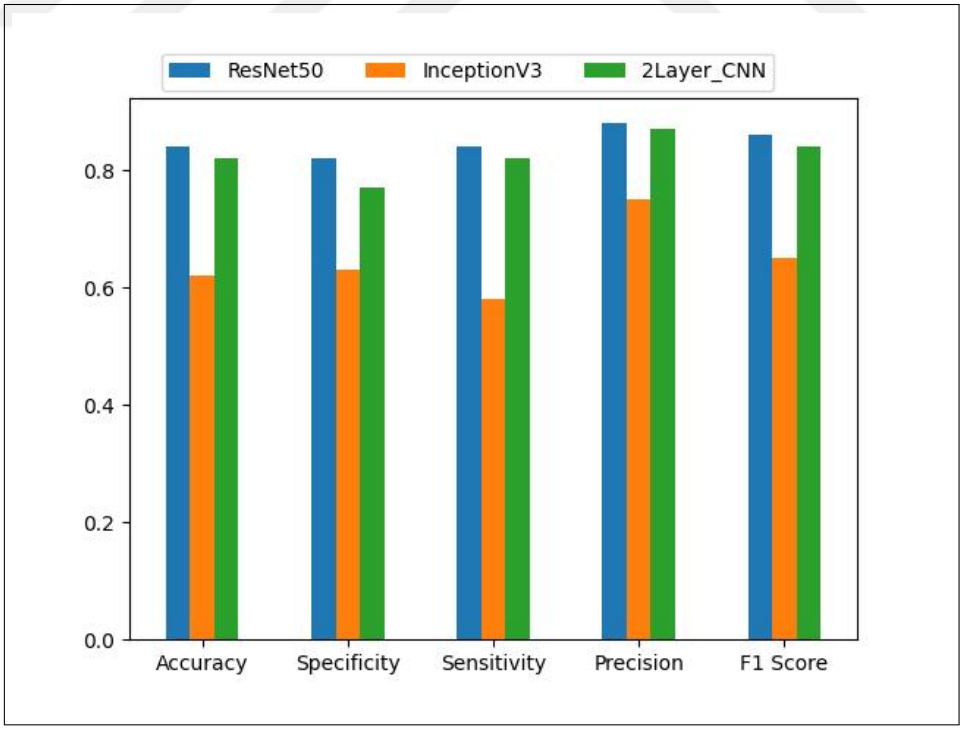



Figure 6.3. The comparison of Resnet50, InceptionV3, and 2-layer CNN models with EWC strategy

### 6.1.4. Rehearsal EWC Strategy

The performance metrics for the same models using the Rehearsal EWC strategy are shown in Table 6.4. Resnet50 outperforms other models in all performance metrics for both the Sipakmed and Herlev test sets, with 0.99 accuracy, 1.00 specificity, 0.98 sensitivity, 1.00 precision, and 0.99 f1 score for Sipakmed and 0.98 accuracy, 0.94 specificity, 0.99 sensitivity, 0.98 precision, and 0.99 f1 score for Herlev. Resnet50 outperforms other models on the CRIC test set in terms of accuracy, specificity, precision, and f1 score, with 0.95, 0.94, 0.98, and 0.97, respectively. With 0.97, InceptionV3 has the highest sensitivity.

Table 6.4. Comparative performance metrics of Resnet50, InceptionV3 and 2-layer CNN model with Rehearsal EWC strategy

| Model | Test Dataset | Metric | | | | |
|---|---|---|---|---|---|---|
| | | Accuracy | Specificity | Sensitivity | Precision | F1 Score |
| Resnet50 | Sipakmed | 0.99 | 1.00 | 0.98 | 1.00 | 0.99 |
| | Herlev | 0.98 | 0.94 | 0.99 | 0.98 | 0.99 |
| | CRIC | 0.95 | 0.94 | 0.96 | 0.98 | 0.97 |
| InceptionV3 | Sipakmed | 0.83 | 0.83 | 0.83 | 0.83 | 0.83 |
| | Herlev | 0.77 | 0.49 | 0.87 | 0.82 | 0.84 |
| | CRIC | 0.75 | 0.16 | 0.97 | 0.76 | 0.85 |
| 2-Layer CNN | Sipakmed | 0.91 | 0.86 | 0.96 | 0.87 | 0.91 |
| | Herlev | 0.83 | 0.49 | 0.96 | 0.84 | 0.89 |
| | CRIC | 0.90 | 0.82 | 0.93 | 0.93 | 0.93 |

The Resnet50 has the highest average score for all performance metrics when trained using the Rehearsal EWC strategy, as shown in Figure 6.4. According to McNemar's test, Resnet50 outperforms the InceptionV3 and 2-layer CNN models ($p<0.05$), while the 2-layer CNN model outperforms the InceptionV3 model ($p<0.05$).

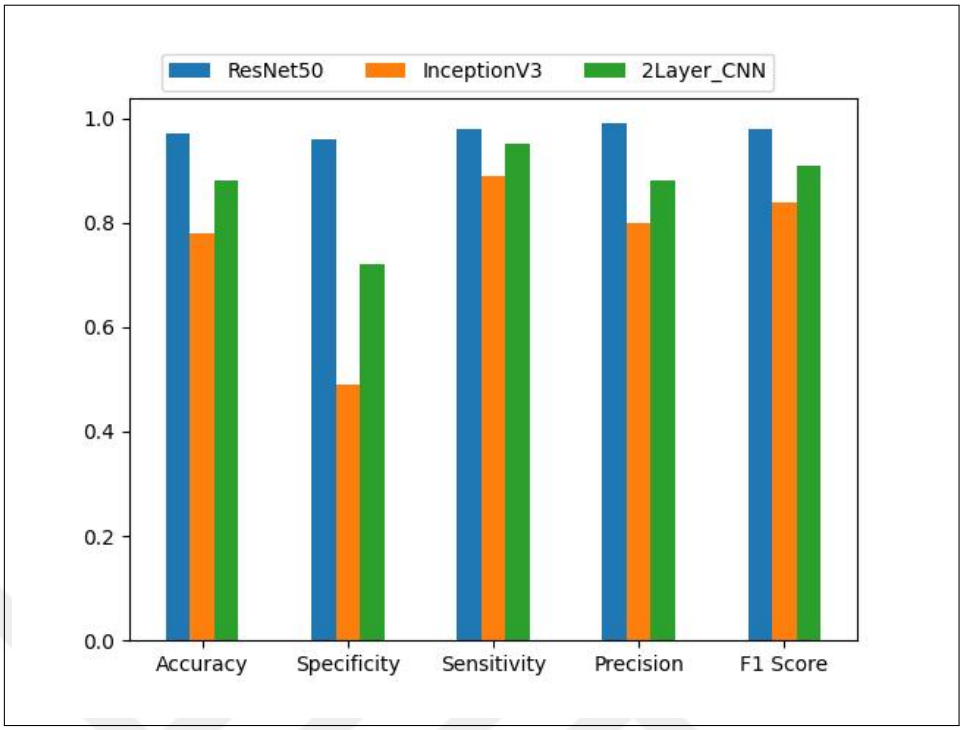



Figure 6.4. The comparison of Resnet50, InceptionV3, and 2-layer CNN models with rehearsal-EWC strategy

## 6.2. STRATEGY ANALYSIS

The strategy analysis contrasts the performance of CNN models (Resnet50, InceptionV3, and 2-layer CNN) with the various continual learning strategies proposed in this study. The continuous learning strategies are naive, rehearsal, EWC, and rehearsal EWC. In the following sections, the effects of strategies are examined separately among the CNN models.

### 6.2.1. Resnet50 Model

Table 6.5 displays the accuracy, specificity, sensitivity, precision, and f1-score measures obtained from the Naive, Rehearsal, EWC, and Rehearsal EWC strategies using the Resnet50 model. Rehearsal EWC outperforms all other strategies in all performance metrics for all test sets, with 0.99 accuracy, 1.00 specificity, 0.98 sensitivity, 1.00 precision, and 0.99 f1 score for Sipakmed; 0.98 accuracy, 0.94 specificity, 0.99 sensitivity, 0.98 precision, and 0.99 f1 score for Herlev; 0.95 accuracy, 0.94 specificity, 0.96 sensitivity, 0.98 precision, and 0.97 f1 score for CRIC test set.

Table 6.5. Comparative performance metrics of Naive, Rehearsal, EWC, and Rehearsal EWC strategies with Resnet50 model

| Strategy | Test Dataset | Metric | | | | |
|---|---|---|---|---|---|---|
| | | Accuracy | Specificity | Sensitivity | Precision | F1 Score |
| Naive | Sipakmed | 0.63 | 0.79 | 0.48 | 0.70 | 0.57 |
| | Herlev | 0.78 | 0.88 | 0.75 | 0.94 | 0.83 |
| | CRIC | 0.97 | 0.93 | 0.98 | 0.98 | 0.98 |
| Rehearsal | Sipakmed | 0.92 | 0.90 | 0.95 | 0.90 | 0.93 |
| | Herlev | 0.80 | 0.67 | 0.84 | 0.88 | 0.86 |
| | CRIC | 0.89 | 0.84 | 0.91 | 0.94 | 0.92 |
| EWC | Sipakmed | 0.72 | 0.78 | 0.66 | 0.75 | 0.70 |
| | Herlev | 0.85 | 0.73 | 0.89 | 0.90 | 0.90 |
| | CRIC | 0.96 | 0.95 | 0.97 | 0.98 | 0.97 |
| Rehersal EWC | Sipakmed | 0.99 | 1.00 | 0.98 | 1.00 | 0.99 |
| | Herlev | 0.98 | 0.94 | 0.99 | 0.98 | 0.99 |
| | CRIC | 0.95 | 0.94 | 0.96 | 0.98 | 0.97 |

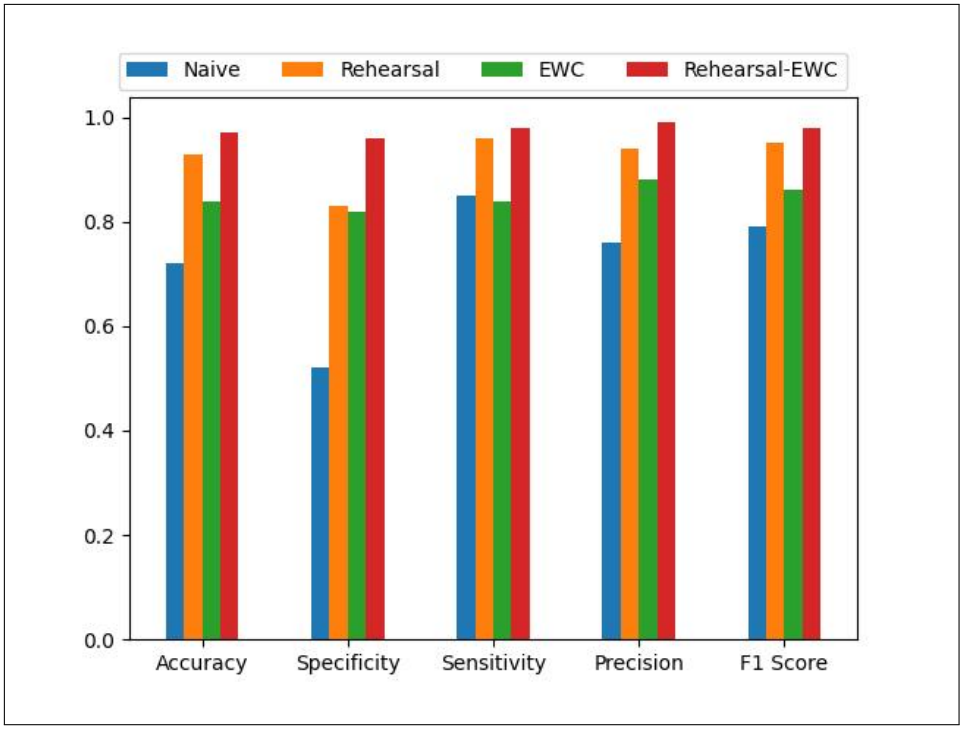



Figure 6.5. The comparison of naive, rehearsal, EWC, and rehearsal-EWC strategies with Resnet50 model

The Rehearsal EWC strategy has the highest average score for all performance metrics when trained using the Resnet50 architecture, as shown in Figure 6.5. According to McNemar's test, Rehearsal EWC outperforms all other strategies ($p<0.05$), rehearsal strategy outperforms both naive and EWC strategies ($p<0.05$), and there is no statistical difference between the naive and EWC strategies ($p=0.27$).

### 6.2.2. InceptionV3 Model

Table 6.6. Comparative performance metrics of Naive, Rehearsal, EWC, and Rehearsal EWC strategies with InceptionV3 model

| Strategy | Test Dataset | Metric | | | | |
|---|---|---|---|---|---|---|
| | | Accuracy | Specificity | Sensitivity | Precision | F1 Score |
| Naive | Sipakmed | 0.62 | 0.86 | 0.38 | 0.74 | 0.51 |
| | Herlev | 0.58 | 0.80 | 0.50 | 0.87 | 0.63 |
| | CRIC | 0.84 | 0.79 | 0.86 | 0.92 | 0.89 |
| Rehearsal | Sipakmed | 0.82 | 0.71 | 0.92 | 0.76 | 0.84 |
| | Herlev | 0.85 | 0.57 | 0.96 | 0.86 | 0.91 |
| | CRIC | 0.81 | 0.65 | 0.87 | 0.87 | 0.87 |
| EWC | Sipakmed | 0.58 | 0.71 | 0.45 | 0.61 | 0.52 |
| | Herlev | 0.56 | 0.69 | 0.51 | 0.82 | 0.63 |
| | CRIC | 0.71 | 0.49 | 0.79 | 0.81 | 0.80 |
| Rehersal EWC | Sipakmed | 0.83 | 0.83 | 0.83 | 0.83 | 0.83 |
| | Herlev | 0.77 | 0.49 | 0.87 | 0.82 | 0.84 |
| | CRIC | 0.75 | 0.16 | 0.97 | 0.76 | 0.85 |

Table 6.6 shows the performance metrics for the same strategies with the InceptionV3 architecture. Regarding accuracy and precision for the Sipakmed test set, the Rehearsal EWC strategy outperforms the other strategies, with 0.83 for both metrics. However, regarding sensitivity and f1 score, the Rehearsal strategy outperforms the other strategies with 0.92 and 0.84, respectively. With a score of 0.86, the naive strategy performs best in terms of specificity. The Rehearsal strategy outperforms the other strategies in terms of accuracy, sensitivity, and f1 score for the Herlev test set, with 0.85, 0.96, and 0.91, respectively. However, the Naive strategy outperforms the other strategies in specificity and precision, with 0.80 and 0.87, respectively. For the CRIC test set, the naive strategy outperforms the

other strategies in terms of accuracy, specificity, precision, and f1 score, with 0.84, 0.79, 0.92, and 0.89, respectively. However, in terms of sensitivity, the Rehearsal EWC strategy outperforms other strategies with a score of 0.97.

When trained using the InceptionV3 architecture, the Rehearsal strategy has the highest average score for all performance metrics, as shown in Figure 6.6. All other strategies are outperformed by the rehearsal strategy ($p<0.05$), the rehearsal EWC strategy outperforms both the naive and EWC strategies ($p<0.05$), and the naive strategy outperforms the EWC strategy ($p<0.05$).

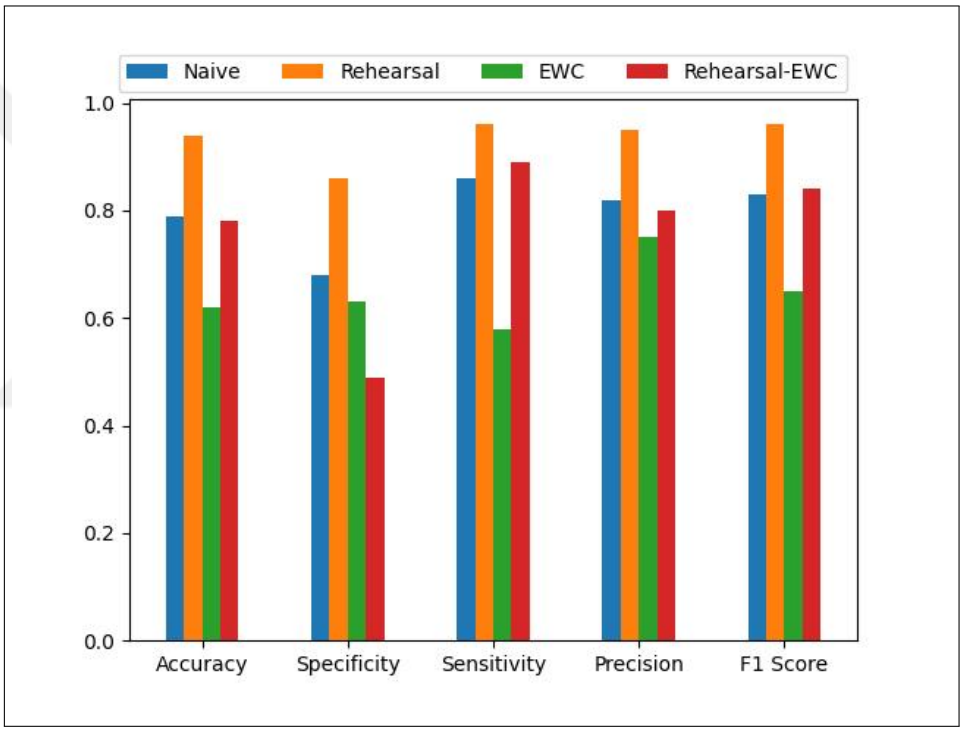



Figure 6.6. The comparison of naive, rehearsal, EWC, and rehearsal-EWC strategies with InceptionV3 model

### 6.2.3. 2-Layer CNN Model

The performance metrics for the same strategies with the 2-layer CNN architecture are shown in Table 6.7. With 0.95 accuracy, 0.94 specificity, 0.97 sensitivity, 0.94 precision, and 0.95 f1 score, the Rehearsal strategy outperforms all other strategies in all performance metrics for the Sipakmed test set. Rehearsal strategy outperforms other strategies in Herlev test set in accuracy, precision, and f1 score with 0.86, 0.87, and 0.91, respectively; it shares the best

Table 6.7. Comparative performance metrics of Naive, Rehearsal, EWC, and Rehearsal EWC strategies with 2-layer CNN model

| Strategy | Test Dataset | Metric | | | | |
|---|---|---|---|---|---|---|
| | | Accuracy | Specificity | Sensitivity | Precision | F1 Score |
| Naive | Sipakmed | 0.78 | 0.74 | 0.82 | 0.76 | 0.79 |
| | Herlev | 0.77 | 0.49 | 0.87 | 0.82 | 0.84 |
| | CRIC | 0.94 | 0.87 | 0.96 | 0.95 | 0.96 |
| Rehearsal | Sipakmed | 0.95 | 0.94 | 0.97 | 0.94 | 0.95 |
| | Herlev | 0.86 | 0.59 | 0.96 | 0.87 | 0.91 |
| | CRIC | 0.92 | 0.82 | 0.95 | 0.94 | 0.94 |
| EWC | Sipakmed | 0.76 | 0.85 | 0.68 | 0.82 | 0.74 |
| | Herlev | 0.79 | 0.59 | 0.87 | 0.85 | 0.86 |
| | CRIC | 0.90 | 0.86 | 0.91 | 0.95 | 0.93 |
| Rehersal EWC | Sipakmed | 0.91 | 0.86 | 0.96 | 0.87 | 0.91 |
| | Herlev | 0.83 | 0.49 | 0.96 | 0.84 | 0.89 |
| | CRIC | 0.90 | 0.82 | 0.93 | 0.93 | 0.93 |

score with EWC strategy in specificity with 0.59, and Rehearsal EWC strategy in sensitivity with 0.96. Finally, for the CRIC test set, the Naive strategy outperforms other strategies in terms of accuracy, specificity, sensitivity, and f1 score, with 0.94, 0.87, 0.96, and 0.96, respectively; it shares the best precision score with EWC strategy, with 0.95.

The Rehearsal strategy has the highest average score for all performance metrics when trained using the 2-layer CNN architecture, as shown in Figure 6.7. McNemar's test shows that the Rehearsal strategy outperforms all other strategies ($p<0.05$), the Rehearsal EWC strategy outperforms both the Naive and EWC strategies ($p<0.05$), and the Naive strategy outperforms the EWC strategy ($p<0.05$).

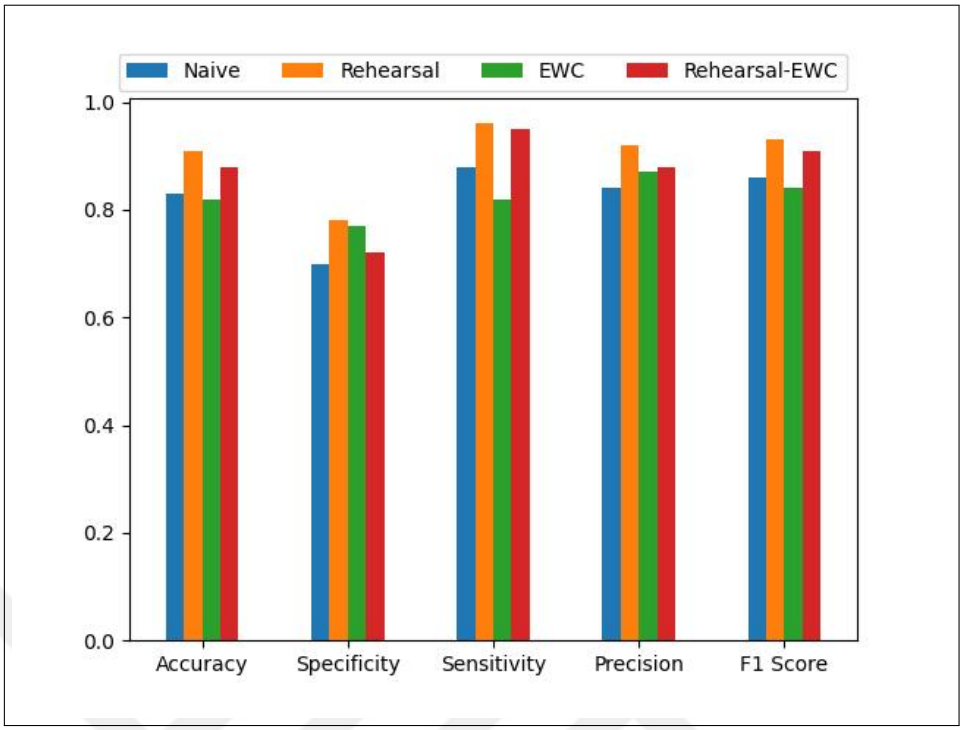



Figure 6.7. The comparison of naive, rehearsal, EWC, and rehearsal-EWC strategies with 2-layer CNN model

# 7. CONCLUSION

This study examines that catastrophic forgetting is inevitable when a deep learning model is trained on diverse pap smear datasets, causing dataset shift in a sequential manner. In order to avoid or at least minimize the impact of catastrophic forgetting, continual learning strategies are applied to the model training, which has already been proven their success on different image processing benchmark datasets in various domains.

Two popular CNN architectures, ResNet50 and InceptionV3, and a custom CNN model have been selected to build the model, apply the continual learning strategies, and run the experiments. The models are trained and tested on three benchmark pap smear datasets: Herlev, Sipakmed, and CRIC.

As continual learning strategies, naive, rehearsal, EWC, and combination of rehearsal EWC are employed. Regarding success in discrimination of cell categories, rehearsal strategy combined with EWC is the best for ResNet50, while rehearsal strategy is the best for InceptionV3 architecture and the custom 2-layer CNN model. However, they require more memory with every new dataset and new training phase.

Naive strategy is the worst scenario; in other words, the base case of such a task and the rehearsal strategy is the best scenario in terms of accuracy but the worst scenario for memory requirement. The aim of finding and applying continual learning strategies is to find a solution that achieves competitive accuracy with rehearsal strategy more efficiently.

# REFERENCES


1. Bray F, Ferlay J, Soerjomataram I, Siegel RL, Torre LA, Jemal A. Global cancer statistics 2018: GLOBOCAN estimates of incidence and mortality worldwide for 36 cancers in 185 countries. *CA: a cancer journal for clinicians*. 2018;68(6):394-424.

2. Elovainio L, Nieminen P, Miller A. Impact of cancer screening on women's health. *International Journal of Gynecology & Obstetrics*. 1997;58(1):137-47.

3. Elsheikh TM, Austin RM, Chhieng DF, Miller FS, Moriarty AT, Renshaw AA. American society of cytopathology workload recommendations for automated pap test screening: Developed by the productivity and quality assurance in the era of automated screening task force. *Diagnostic cytopathology*. 2013;41(2):174-8.

4. Lozano R. Comparison of computer-assisted and manual screening of cervical cytology. *Gynecologic oncology*. 2007;104(1):134-8.

5. Goodfellow I, Bengio Y, Courville A. Deep learning. Book in preparation for MIT Press. *URL¡ http://www deeplearningbook org*. 2016;1.

6. LeCun Y, Bengio Y, Hinton G. Deep learning. *nature*. 2015;521(7553):436-44.

7. Xing F, Yang L. Robust nucleus/cell detection and segmentation in digital pathology and microscopy images: a comprehensive review. *IEEE reviews in biomedical engineering*. 2016;9:234-63.

8. Raghu M, Zhang C, Kleinberg J, Bengio S. Transfusion: Understanding transfer learning for medical imaging. *Advances in neural information processing systems*. 2019;32.

9. Pianykh OS, Langs G, Dewey M, Enzmann DR, Herold CJ, Schoenberg SO, et al. Continuous learning AI in radiology: implementation principles and early applications. *Radiology*. 2020;297(1):6-14.

10. Castro DC, Walker I, Glocker B. Causality matters in medical imaging. *Nature Communications*. 2020;11(1):1-10.

11. Moreno-Torres JG, Raeder T, Alaiz-Rodríguez R, Chawla NV, Herrera F. A unifying

view on dataset shift in classification. *Pattern recognition*. 2012;45(1):521-30.

12. Parisi GI, Kemker R, Part JL, Kanan C, Wermter S. Continual lifelong learning with neural networks: A review. *Neural Networks*. 2019;113:54-71.

13. De Lange M, Aljundi R, Masana M, Parisot S, Jia X, Leonardis A, et al. A continual learning survey: Defying forgetting in classification tasks. *IEEE transactions on pattern analysis and machine intelligence*. 2021;44(7):3366-85.

14. McCloskey M, Cohen NJ. Catastrophic interference in connectionist networks: The sequential learning problem. *Psychology of learning and motivation*. vol. 24. Elsevier. 1989:109-65.

15. Bengtsson E, Malm P. Screening for cervical cancer using automated analysis of PAP-smears. *Computational and mathematical methods in medicine*. 2014;2014.

16. Zhang L, Kong H, Ting Chin C, Liu S, Fan X, Wang T, et al. Automation-assisted cervical cancer screening in manual liquid-based cytology with hematoxylin and eosin staining. *Cytometry Part A*. 2014;85(3):214-30.

17. Win KY, Choomchuay S, Hamamoto K, Raveesunthornkiat M, Rangsirattanakul L, Pongsawat S. Computer aided diagnosis system for detection of cancer cells on cytological pleural effusion images. *BioMed research international*. 2018;2018.

18. Zhang L, Lu L, Nogues I, Summers RM, Liu S, Yao J. DeepPap: deep convolutional networks for cervical cell classification. *IEEE journal of biomedical and health informatics*. 2017;21(6):1633-43.

19. Lin H, Hu Y, Chen S, Yao J, Zhang L. Fine-grained classification of cervical cells using morphological and appearance based convolutional neural networks. *IEEE Access*. 2019;7:71541-9.

20. Nguyen LD, Gao R, Lin D, Lin Z. Biomedical image classification based on a feature concatenation and ensemble of deep CNNs. *Journal of Ambient Intelligence and Humanized Computing*. 2019:1-13.

21. Rahaman MM, Li C, Yao Y, Kulwa F, Wu X, Li X, et al. DeepCervix: a deep

learning-based framework for the classification of cervical cells using hybrid deep feature fusion techniques. *Computers in Biology and Medicine*. 2021;136:104649.

22. Manna A, Kundu R, Kaplun D, Sinitca A, Sarkar R. A fuzzy rank-based ensemble of CNN models for classification of cervical cytology. *Scientific Reports*. 2021;11(1):1-18.

23. Pramanik R, Biswas M, Sen S, de Souza Júnior LA, Papa JP, Sarkar R. A fuzzy distance-based ensemble of deep models for cervical cancer detection. *Computer Methods and Programs in Biomedicine*. 2022;219:106776.

24. Perkonigg M, Hofmanninger J, Herold CJ, Brink JA, Pianykh O, Prosch H, et al. Dynamic memory to alleviate catastrophic forgetting in continual learning with medical imaging. *Nature Communications*. 2021;12(1):1-12.

25. Shorten C, Khoshgoftaar TM. A survey on image data augmentation for deep learning. *Journal of big data*. 2019;6(1):1-48.

26. Taylor L, Nitschke G. Improving deep learning with generic data augmentation. *2018 IEEE Symposium Series on Computational Intelligence (SSCI)*. IEEE. 2018:1542-7.

27. Canny J. A computational approach to edge detection. *IEEE Transactions on pattern analysis and machine intelligence*. 1986;PAMI-8(6):679-98.

28. Ratcliff R. Connectionist models of recognition memory: constraints imposed by learning and forgetting functions. *Psychological review*. 1990;97(2):285.

29. Kirkpatrick J, Pascanu R, Rabinowitz N, Veness J, Desjardins G, Rusu AA, et al. Overcoming catastrophic forgetting in neural networks. *Proceedings of the national academy of sciences*. 2017;114(13):3521-6.

30. Plissiti ME, Dimitrakopoulos P, Sfikas G, Nikou C, Krikoni O, Charchanti A. SIPAKMED: A new dataset for feature and image based classification of normal and pathological cervical cells in Pap smear images. *2018 25th IEEE International Conference on Image Processing (ICIP)*. IEEE. 2018:3144-8.

31. Jantzen J, Norup J, Dounias G, Bjerregaard B. Pap-smear benchmark data for pattern classification. *Nature inspired Smart Information Systems (NiSIS 2005)*. 2005:1-9.

32. Basak H, Kundu R, Chakraborty S, Das N. Cervical cytology classification using PCA and GWO enhanced deep features selection. *SN Computer Science*. 2021;2(5):1-17.

33. Rezende MT, Silva R, Bernardo FdO, Tobias AH, Oliveira PH, Machado TM, et al. Cric searchable image database as a public platform for conventional pap smear cytology data. *Scientific Data*. 2021;8(1):1-8.

34. N Diniz D, T Rezende M, GC Bianchi A, M Carneiro C, JS Luz E, JP Moreira G, et al. A deep learning ensemble method to assist cytopathologists in pap test image classification. *Journal of Imaging*. 2021;7(7):111.

35. McNemar Q. Note on the sampling error of the difference between correlated proportions or percentages. *Psychometrika*. 1947;12(2):153-7.