ISTANBUL TECHNICAL UNIVERSITY
ELECTRICAL ELECTRONICS FACULTY

NEURAL NETWORK DESIGN FOR SOLVING THE DATA CLASS IMBALANCE PROBLEM AND ITS APPLICATIONS ON MEDICAL DATASETS

SENIOR DESIGN PROJECT
INTERIM REPORT

Emirhan Bilgiç
040190006

Emre Akcin
040180038

Project Advisor: Assc. Prof. İsa Yıldırım

ELECTRONICS AND COMMUNICATION ENGINEERING DEPARTMENT

JUNE, 2023

We are submitting the Senior Design Project Interim Report entitled as “NEURAL NETWORK DESIGN FOR SOLVING THE DATA CLASS IMBALANCE PROBLEM AND ITS APPLICATIONS ON MEDICAL DATASETS”. The Senior Design Project Interim Report has been prepared as to fulfill the relevant regulations of the Electronics and Communication Engineering Department of Istanbul Technical University. We hereby confirm that we have realized all stages of the Senior Design Project Interim Report by ourselves, and we have abided by the ethical rules.

Name SURNAME Emirhan Bilgiç
040190006

Name SURNAME Emre Akcin
040180038

FOREWORD

We would like to thank Assoc. Prof. Dr. İsa Yıldırım and Prof. Dr. Neslihan Serap Şengör and our esteemed families for their help throughout our final thesis.

June 2023

Emirhan Bilgiç
Emre Akcin

TABLE OF CONTENTS

Page

FOREWORD iii

TABLE OF CONTENTS iv

SUMMARY v

1. INTRODUCTION

1.1 Project Proposal 6

1.2 Suggested Work Plan and Possible Changes in the Project Proposal .. 6, 7

1.3 Literature Review 8

2. PLANNED AND REALIZED STAGES IN THE PROJECT PROPOSAL

2.1 Description of the Problem 9

2.2 Concrete Examples of the Problem 9, 10

2.3 Purpose of the Project 11

2.4 Approach to the Project and Requirements 12

2.5 Impacts of the Project 11

2.6 Plans and Method in the Project 12-18

2.7 Realization of the Project 18-22

2.8 Recommendations for the Future 22

REFERENCES 23

NEURAL NETWORK DESIGN FOR SOLVING THE DATA CLASS IMBALANCE PROBLEM AND ITS APPLICATIONS ON MEDICAL DATASETS

SUMMARY

This project focuses on a common problem in deep learning and machine learning applications, namely the problem of data class imbalance. This problem is particularly common in multiclass and binary classification problems, and is often seen in medical datasets. Artificial intelligence models trained with data sets with data imbalance problem have a bias towards the majority class when making predictions, which leads to incorrect predictions.

Data imbalance negatively affects the training process of the model and reduces the overall success of the model. This usually occurs when one class in the dataset has more data than the other classes. For example, when considering an Alzheimer's disease detection model, there may be significant differences between the amount of data represented by different stages of the disease. This may cause the model to over-learn some classes or under-learn others. For example, if the amount of MRI images with mild dementia is greater than the amount of MRI images with severe dementia, the model may classify each new sample as mild dementia, which would negatively affect the performance of the model.

In the project, a more up-to-date and practically usable model/technique was developed, taking into account the unbalanced data distribution. For this purpose, existing solutions and their weaknesses were analyzed and in the light of this information, it was aimed to create a more successful technique.

One of the main objectives is to automate some of the steps that are done manually and to achieve efficiency in terms of both time and labor. In the project, besides the techniques used in data imbalance problems and available in the literature (such as SMOTEENN), original algorithms were tested. It is aimed to use the most appropriate of these techniques according to different data set scenarios.

In addition, this project also aims to propose solutions to improve the initial parameters of the model. Thus, it is aimed to improve the overall performance of the model.

As a result, by solving the data imbalance problem and optimizing the model parameters, this project has enabled machine learning and deep learning applications to be more successful. A combined algorithm was written using both existing techniques and original methods. In the analysis, it was determined that the models trained with the algorithm developed within the scope of this project were more successful compared to the models trained with data sets with data imbalance problem.

1. INTRODUCTION

1.1 Project Proposal

The project focuses on the imbalanced classes (data class imbalance) problem encountered in deep learning and machine learning problems, especially in multiclass and binary classification problems. The data imbalance problem is the name given to the uneven distribution in data classes. This situation is especially encountered in medical data sets. The solution proposal is to solve the problem of irregular data clustering and random parameters in the deep learning model to be used in the main problem with the model to be developed. By focusing on the existing solutions in the literature and their weaknesses, it is aimed to create this more up-to- date and practical model with the contribution of existing studies.

1.2 Suggested Work Plan and Possible Changes in the Project Proposal

It may be encountered that the model structure implemented in the project reduces the current model's success. This situation can arise from insufficient mathematics and software foundations. In order to prevent such risks, first of all, the project will be based on a solid mathematical and software foundation. Current literature will be scanned in detail. In addition,repetitive checks and a self-improving feedback structure will be established so that the modeldoes not make wrong decisions. The established structure will definitely be tested with real data.

Possible changes; These can be listed as switching to a different method in case the written neural network structure reduces the current success, trying a different structure in case the feedback mechanism does not work properly or removing the feedback mechanism completely.

In general terms, the group work plan is as in the table. (Table 1.1)

#	MISSIONS	RESPONSIBLE GROUP MEMBER (MEMBERS)	WEEKS																												
			1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28
1	Determining whether there is a imbalanced classes problem in the data set	Emre Akcin, Emirhan Bilgiç,				X	X	X	X	X	X																			
2	Determination of optimum values of other parameters	Emre Akcin, Emirhan Bilgiç								X	X	X	X	X	X	X														
3	In case of problem, determining the required number of data numbers in data classes	Emre Akcin, Emirhan Bilgiç												X	X	X	X	X	X	X	X	X	X							
4	Automatic adjustment of the data set according to CNN's prediction results	Emre Akcin, Emirhan Bilgiç																	X	X	X	X	X	X	X	X				
5	Identification and comparison of CNN's first and last success	Emre Akcin																								X	X			
6	Comparing the success of the neural network with the success of the currently used methods	Emirhan Bilgiç																										X	X	X

Table 1.1: Group Work Plan

1.3 Literature Research

One of the most well-known parameter optimization techniques in the literature is the Python library called Hyperopt. Enabling model-based optimization, also called Bayesian optimization, it leads to a more efficient training process, thereby increasing accuracy and othermetric success. Hyperopt library provides algorithms and a parallelization engine to perform hyperparameter optimization (model selection) in Python. [1]

On the other hand, there is the SMOTE technique, which explains the balance between classesby reducing it to a formulation, especially in order to prevent imbalance in tabular data. This method is also used in the literature and its success has been proven by testing on methods suchas the Naive Bayes classifier [2]. On the other hand, it is a shortcoming aspect of this techniquethat it has not been tried on non-table data such as images, and therefore has not proven its success.

Another technique used in the problem of unbalanced datasets in the literature is the data augmentation method, which is very popular. This method is not automated but consists of randomly increasing the data number of the class with relatively little data. [3] Although it increases the number of data, it is not automated and the absence of an algorithm that decides the numbers are the serious shortcomings and weaknesses of the method.

In addition, the method called Transfer Learning has been found to be useful in cases of data scarcity and imbalance. [4] In this method, a Convolutional Neural Network (CNN) previouslytrained with a different dataset is used. With the integration of CNN into a specific dataset, the problem of insufficient and unbalanced data in the dataset can be partially avoided. This method is generally used in cases where there is little data, and it is not sufficient to solve the problem of unbalanced datasets.

2. PLANNED AND REALIZED STAGES IN THE PROJECT PROPOSAL

2.1 Description of the Problem

The project focuses on the imbalanced classes problem, which is frequently encountered in deep learning and machine learning problems, especially in classification problems. When trying to teach such a data set to a model, overfitting can be observed in a class with a large number of data, or underfitting in a class with a small number of data. The data imbalance problem causes the training process to be adversely affected and the model success/accuracy to decrease significantly.

2.2 Concrete Examples of the Problem

The imbalanced classes problem arises especially in binary and multiple classification problems. To give an example of such problems:

- Fraud Detection
- Demand Forecast
- Spam Detection
- Anomaly Detection

Examples of datasets with this kind of data imbalance are medical datasets. For example, assuming that the goal of a deep learning model is to classify the stages of Alzheimer's disease and the data set consisting of MRI images is distributed as follows:

- Healthy Individual (1500 MR images)
- Mild (Early) Alzheimer's (600 MR images)
- Moderate Alzheimer's (1400 MR images)
- Severe (Late) Alzheimer's (4500 MR images)

If the above dataset is given to train the model and the model is not resistant to the data imbalance problem, it may cause it to fail to learn the mild Alzheimer's class at all or to

overlearn the severe Alzheimer's class. This causes the model to predict the image as this classwhen given an MR image that is not in the training set, even though it is not a severe Alzheimer's image due to overfitting.

2.3 Purpose of the Project

The imbalanced classes problem mentioned above is the main problem for which the project proposes a solution. However, in this project, another common problem, which is the training of the model to be trained with random parameters, and the inability to get the optimum efficiency that can potentially be obtained from a model, will also be solved.

As a solution proposal, the model to be developed and a model to be trained with a data set with data imbalance are both improving the initial parameters and ensuring that the training data is adjusted in a way that does not create data imbalance. With this solution proposal, it isplanned to establish a more efficient link between the model and the training data and to obtain the optimum efficiency that can be obtained from the model by adjusting its parameters.

2.4 Approach to the Project and Requirements

There are solutions for the problems mentioned in the literature and in practice. However, with this project, it is planned to focus on the missing points of older literature studies and to progress by building on these solution proposals. In this way, it is thought to create a more up-to-date and more practical design with the contributions of previous studies. For this reason, we also focused on existing literature studies.

Since our project will work with current models and data sets and will support current technological and scientific studies, the model in the solution proposal has been supported with current libraries and software.

Since its main purpose is to accelerate the process and increase the efficiency of other projects, a highly scalable model has been considered. In this way, the work of other software developers and researchers who want to produce solutions to certain classification problems is accelerated and facilitated.

Software and requirements used in the realization of the project:

- Python programming language

- Anaconda software package
- Data editing libraries such as Numpy, Pandas, etc.
- Tensorflow, Keras and Pytorch libraries for the creation of CNN
- Scikit-learn library for optimization and model results
- Matplotlib and Seaborn libraries for output visualization

Hardware requirements include a computer capable of training our model and a fast internet connection.

2.5 Impacts of the Project

Convolutional Neural Network projects deal with data/data set. The stages of bringing it to the desired state for the model are done manually. With the model created at the end of the project, these processes will be automated. In addition, since training will be realized with more optimum parameters, the performance of the projects to be developed, i.e. the success rate, has increased. In this way, projects developed for the benefit of society will be faster and more successful. This has a positive impact both socially and economically. It saves both energy and time.

2.6 Plans and Method in the Project

The method to be used in the project starts with determining whether there is a data imbalance problem by evaluating the number of data on the basis of class in the data set to beused in the classification problem and the estimation results given by the CNN who passed the training.

CNN is a deep learning model used in image recognition [5]. In the figure below (Figure 2.1),an example of CNN is used to solve the 10-class classification problem of a given image.

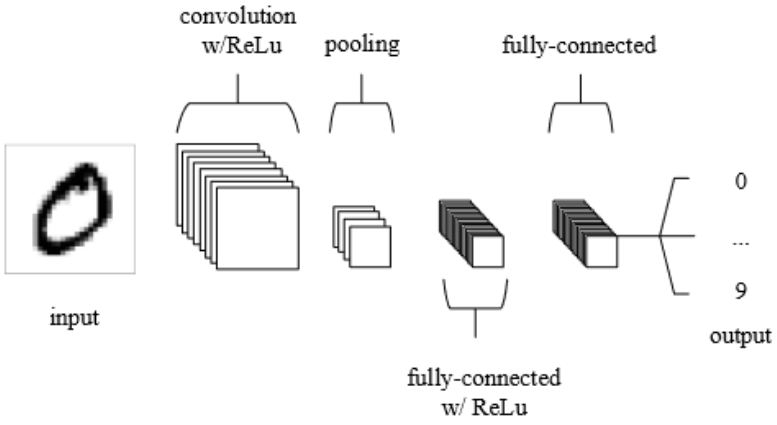


Figure 2.1: Convolutional Neural Network Example [5]

In order to compare the change in success as a result of the modifications, a comparison of several metrics used to determine the performance of the model will be made. Although Accuracy is the most commonly used metric to measure the performance of the model, it maynot be reliable in a model trained with a data set that has data imbalance.

For example, let's consider a dataset with a total of 150 samples and two classes. When class-0 (the weighted class) has 138 instances and class-1 (the minority class) has 12 instances, thisdata set is imbalanced. When we look at the error (complexity) matrix of a model trained withthis dataset; TN (True Negative) = 137, TP (True Positive) = 1, FP (False Positive) = 1 and FN(False Negative) = 11, we see that although the accuracy is 92%, it correctly classified only 1 out of 12 data belonging to the minority class. Therefore, other metrics such as precision, recalland f1-score should be used to see if there is a data imbalance problem in the dataset. In orderto look at these metrics, the model needs to be trained once with the raw data, but if a clear imbalance is detected when looking at the number of data in the classes before training the model, the next step can be taken.

For example, class-0 has 1500 instances while class-1 has only 50 instances.

In the next step, if it is decided that there is an unbalanced dataset, resampling is performed. There are two actions for this:

- Oversampling: Increasing the minority class data with methods such as SMOTEand ADASYN [6].
- Under-sampling (Undersampling): Reducing the data belonging to the weightedclass with methods such as Random Undersampling, ClusterCentroids and NearMiss (It is not preferred in data sets with little data as it will cause information loss).

ither one of these two actions can be chosen or a combination of these actions, SMOTEENN and SMOTETomek, can be used to balance the data set. In this project, SMOTEENN, which is a combination of oversampling and undersampling, will be used. The SMOTEENN method is a combination of the oversampling technique SMOTE (Synthetic Minority Oversampling Technique) and the undersampling technique ENN (Edited Nearest Neighbor). [7]

Steps of the SMOTE technique:

1. Select a random sample ($\boldsymbol{E_i}$) from the minority class.
2. K nearest neighbors of this sample are identified.
3. The closest of these neighbors is selected $\left(\boldsymbol{E_j}\right)$ and its difference with theoriginal data is taken.
4. This difference is multiplied by a number $(\boldsymbol{\delta})$ between 0 and 1.
5. The value is added to the original data and the new data ($\boldsymbol{E_{new}}$) is added to theminority class.
6. These steps are repeated until the minority class is increased at the desired rate.

If we show the SMOTE technique as a formulation:

$$\boldsymbol{E_{new} = \left(E_i - E_j\right)\delta}$$

Figure 2.2: SMOTE Formulation
[8]

With this technique, the existing data is translated. In order to avoid overlearning or for further oversampling, the algorithm can be applied by randomly selecting from among the K nearest neighbors instead of the closest one. In this way, a more successful technique is applied than the over-sampling method by simply placing duplicates. This provides a resistance to the overlearning problem.

Steps of the ENN technique:

1. A random sample of the weighted class is selected.
2. K nearest neighbors are selected. Usually K=3 is chosen.
3. If the selected class does not belong to the class of the original sample, both the original data and its neighbors are deleted from the weighted class.
4. If the selected class does not belong to the class of the original sample, both the original data and its neighbors are deleted from the weighted class.
5. The steps are repeated until the desired weighted class ratio is achieved.

With SMOTEENN, which is a combination of these two methods, both oversampling and undersampling are used effectively.

After the dataset is organized with the SMOTEENN technique, the model is trained again and the performance metrics mentioned above are checked again. Depending on the status of these performance metrics, one of the following three actions is taken:

1. If the performance is worse than the performance of the model trained with the original data, SMOTEENN is canceled and the following method is applied.
2. If the performance is better than the model trained with the original data but the desired performance is not achieved, SMOTEENN is repeated. However, if there is no significant improvement after 5 iterations, the highest performing model is selected and the model is trained once more by adjusting the class weights in proportion to the imbalance.

The basis of the data imbalance problem is that the model ignores the minority class and favors the weighted class. By adjusting the class weights, the weight of the minority class is increased, thus increasing the overall error and telling the model that the minority class is also important. In this way, the model will now take the minority class into account and higher performance will be achieved.

3. If the model is both more successful and has achieved the desired success, the trainingis terminated.

If the first of these three cases happens, i.e. if the SMOTEENN result shows worse performance, the part of our project that distinguishes it from the literature and makes it original

is applied. This is the automation of manual processes. The general logic and algorithm are listed as follows:

1. The model trained with the original data is given a sample of the test data.
2. If it classifies correctly, no action is taken. But if it misclassifies, the actual class is increased by one by oversampling and the predicted class is decreased by undersampling by the specified amount K.
3. Perform step 2 for all test data and check the model performance.
4. If the model performance is still poor, the above steps are repeated.
5. If the model performance is close for 5 iterations in a row, the process is stopped.

In this way, the solution to the data imbalance problem has been dynamized with feedback methods and automation has been achieved by automating manual deletion-addition methods.

After these stages, the model is fine-tuned by adjusting the hyperparameters using Hyperopt, GridSearch or RandomizedSearchCV methods, which is another goal of the project. This improves the performance of the model [9].

In addition, the data (with original, augmented and deleted labels) is kept in Pandas Dataframes with the number of instances in the classes, making it easier and faster to perform operations and to return to the previous more successful states when the performance of the model

decreases as a result of the operations performed. This will increase the flexibility and manageability of the project.

6. The success of the method used in the project is compared with other methods usedin solving the data imbalance problem.

To summarize and itemize:

- Determining whether there is a data imbalance problem in the data set with performance metrics
- If there is a data imbalance problem, applying the SMOTEENN technique andadjusting the number of data belonging to the classes
- Comparison of the model trained with the original data and the model with the SMOTEENN technique applied
- Applying the actions indicated according to the situations encountered (increasing the actual class of the misclassified class and decreasing the predicted class)
- Determination of optimum values of hyperparameters with methods found in the literature
- Comparison of the performance of the model trained with raw data and the model trained with data to which the above procedures have been applied
- Comparison of the methods currently used and the methods used in the project

The artificial neural network architecture designed in the project and the steps to be implemented can be summarized as follows:

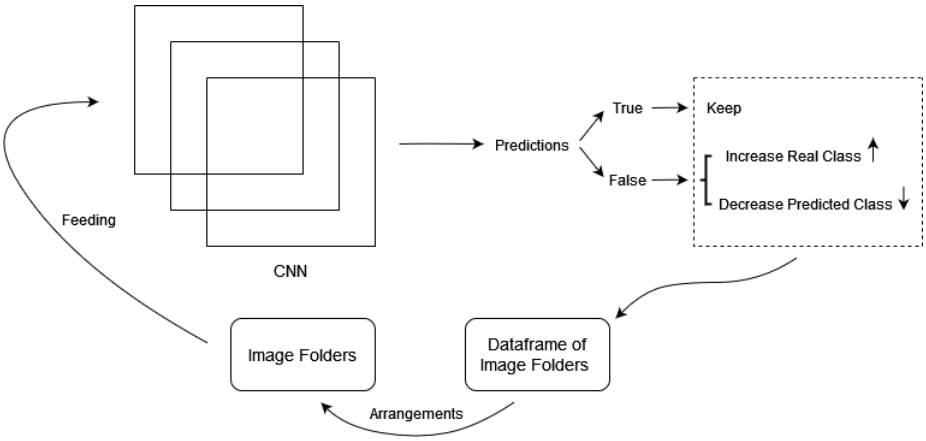


Figure 2.3: Designed Structure in the Project

The programming language to be used in writing the codes in the project is Python, the numpy library will be used to facilitate mathematical operations, the pandas library will be used for data set arrangements, and the Tensorflow, Keras and Pytorch libraries will be used to write the ESA. The scikit learn library can also be used for optimization.

One of the main ethical problems that may arise is the fact that the problem we are trying to solve is especially encountered in medical data sets. The anonymization of medical datasets is meticulously handled. In addition, since reducing the accuracy rate of the classification problem of medical data can have serious consequences in cases such as disease diagnosis, it

must be trained with real data and it must be ensured that the accuracy rate is increased. Otherwise, human health may be jeopardized.

On the other hand, the originality of the project is also an important ethical factor, and in this section, original and unprecedented methods have been tried and compared with other methods in terms of success criteria.

2.7 Project Achievements and Conclusion

Since the project is a software project and an up-to-date project, the necessary libraries and software described in the requirements section were installed.

In addition, methods of detecting the data imbalance problem in data sets were investigated. SMOTE technique was chosen. It was determined that the ratio of the number of data in each data class to other classes should generally be between 3/4 and 4/3.

Hyperopt library was analyzed. It was determined how parameters such as age and number of layers should be organized.

The Learning Transfer method was found to be useful in cases of data scarcity and imbalance. Architectures such as DenseNet-169 and ResNet-50 are compared according to their performance on the Alzheimer's dataset.

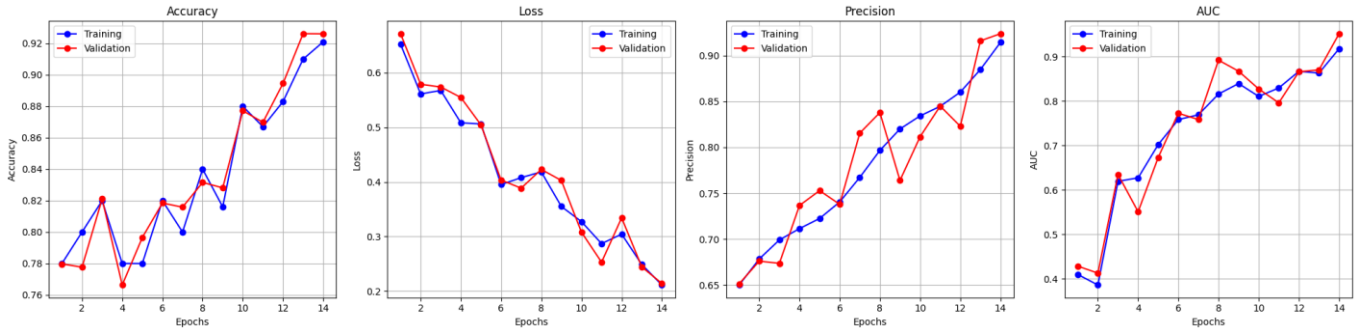
Table 2.1: Comparison of Architectures

Neural Network Architectures	Accuracy	Area Under the Curve
DenseNet-169	%92.10	0.9511
ResNet-50	%71.21	0.8101
ResNet-101	%69.01	0.7220
VGG-16	%68.22	0.7210
VGG-19	%66.72	0.7112

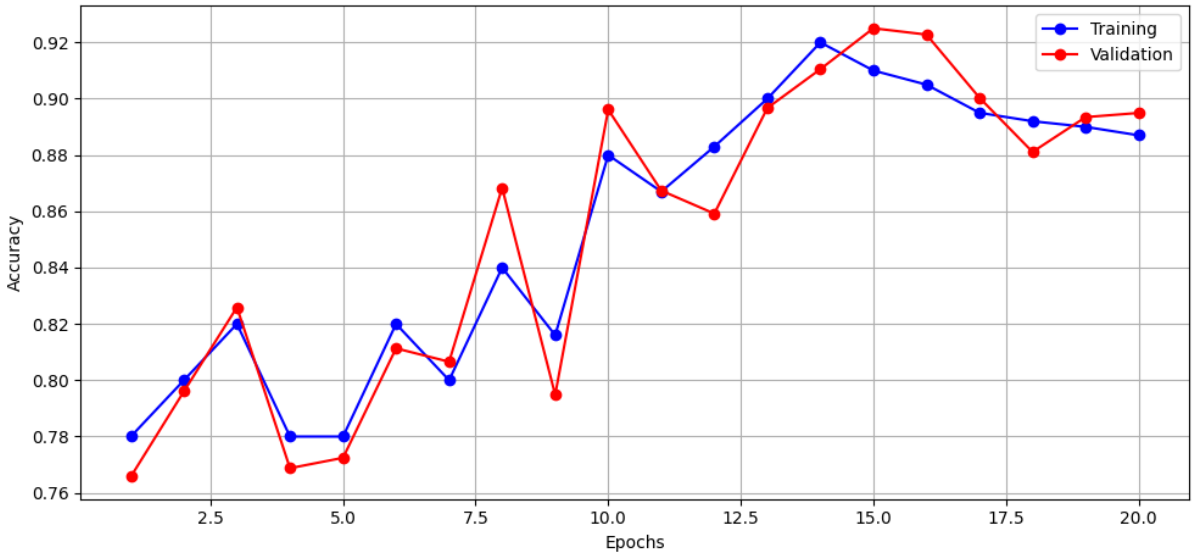
As seen in the table, DenseNet-169 architecture achieved higher success than other architectures. The number of epochs with the highest success was manually targeted as 14. Theaccuracy can be increased by using the DenseNet-169 architecture's ResNet-50 or other

sub-architectures and the results provided by the facilities. This method, known as ensemble learning, is widely used in the literature [10].

The graphs of DenseNet-169 in the training process are as follows (Graph 2.1).



Graph 2.1: Training Process of DenseNet-169



Graph 2.2: DenseNet-169 Accuracy Rate Change

As can be seen from Graph 2.2, the highest accuracy rate is obtained as a result of the 14th epoch.

In the project, neural network training was tried instead of automated data augmentation or reduction methods, but this structure was abandoned due to long training processes and low success rates.

The success of DenseNet-169 was tested on different datasets. The success of the model on the dataset with manual adjustment between classes is much higher than the success with the raw dataset.

On the other hand, the code for the automation of the adjustment process between data classes according to the model's prediction, which was mentioned in the project stages, was written and tested. It was found that this method significantly increases the success rate and saves time.

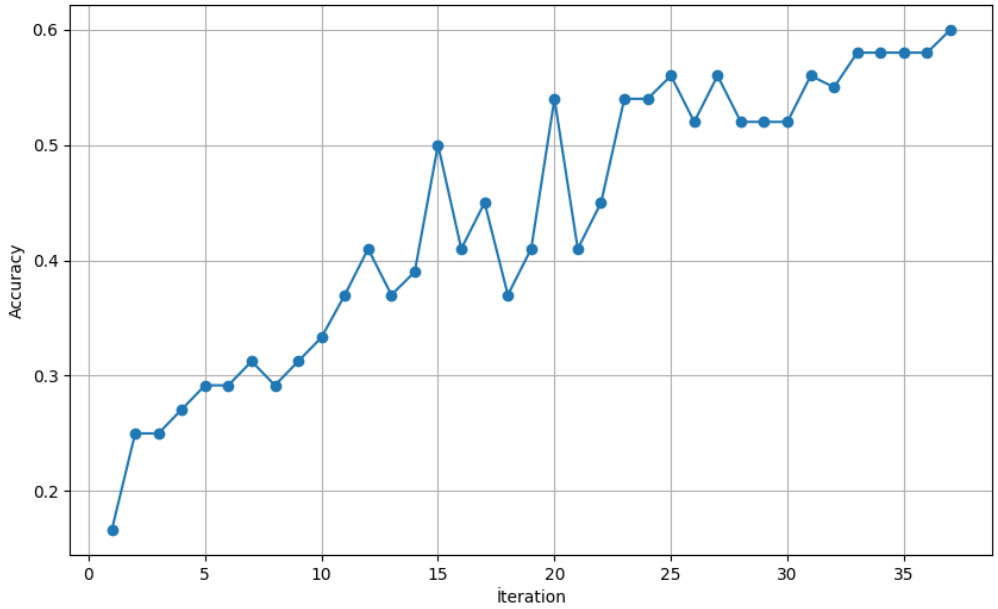
The source code of the project is available in the students' GitHub repository: github.com/emirhanbilgic/SolvingImbalancedClasses-FinalThesis

The automation code written for the realization of the project is based on retraining the model with the updated data set in a while loop. At each iteration, the data set is reorganized with the structure in Figure 2.3 and the training process is started from the beginning. In each iteration, the model is retrained for up to 3 epochs with the edited data set. It was observed that the accuracy of the model converged after a while.

```
Found 238 images belonging to 4 classes.
Epoch 1/3
7/7 [==============================] - 191s 20s/step - loss: 1.2560 - accuracy: 0.4320
Epoch 2/3
7/7 [==============================] - 143s 20s/step - loss: 0.6057 - accuracy: 0.7961
Epoch 3/3
7/7 [==============================] - 147s 21s/step - loss: 0.3546 - accuracy: 0.8883
2/2 [==============================] - 10s 3s/step
Current accuracy: 0.16666666666666666
Predictions: ['ModerateDemented', 'ModerateDemented', 'ModerateDemented', 'ModerateDemented', 'ModerateDemented', 'ModerateDemented', 'ModerateDemented', 'ModerateDemented'
Found 259 images belonging to 4 classes.
Epoch 1/3
8/8 [==============================] - 226s 22s/step - loss: 1.1222 - accuracy: 0.5195
Epoch 2/3
8/8 [==============================] - 159s 19s/step - loss: 0.5498 - accuracy: 0.8150
Epoch 3/3
8/8 [==============================] - 157s 22s/step - loss: 0.3774 - accuracy: 0.8502
WARNING:tensorflow:5 out of the last 9 calls to <function Model.make_predict_function.<locals>.predict_function at 0x7f1653da9e10> triggered tf.function retracing. Tracing
2/2 [==============================] - 9s 2s/step
Current accuracy: 0.25
Predictions: ['ModerateDemented', 'ModerateDemented', 'ModerateDemented', 'ModerateDemented', 'ModerateDemented', 'ModerateDemented', 'ModerateDemented', 'ModerateDemented'
Found 293 images belonging to 4 classes.
Epoch 1/3
9/9 [==============================] - 232s 22s/step - loss: 1.0273 - accuracy: 0.5364
Epoch 2/3
9/9 [==============================] - 177s 20s/step - loss: 0.6074 - accuracy: 0.7816
Epoch 3/3
9/9 [==============================] - 197s 22s/step - loss: 0.3488 - accuracy: 0.8854
WARNING:tensorflow:6 out of the last 11 calls to <function Model.make_predict_function.<locals>.predict_function at 0x7f164c2da560> triggered tf.function retracing. Tracing
2/2 [==============================] - 10s 2s/step
Current accuracy: 0.25
Predictions: ['ModerateDemented', 'ModerateDemented', 'ModerateDemented', 'ModerateDemented', 'ModerateDemented', 'ModerateDemented', 'ModerateDemented', 'ModerateDemented'
Found 317 images belonging to 4 classes.
Epoch 1/3
9/9 [==============================] - 247s 22s/step - loss: 0.9609 - accuracy: 0.5860
Epoch 2/3
9/9 [==============================] - 196s 22s/step - loss: 0.5021 - accuracy: 0.8105
Epoch 3/3
9/9 [==============================] - 196s 22s/step - loss: 0.2391 - accuracy: 0.9263
2/2 [==============================] - 10s 2s/step
```

Figure 2.4: Convergence Output of the Executed Code

When the accuracy of the model exceeded 0.6 after iterations, the process was stopped.



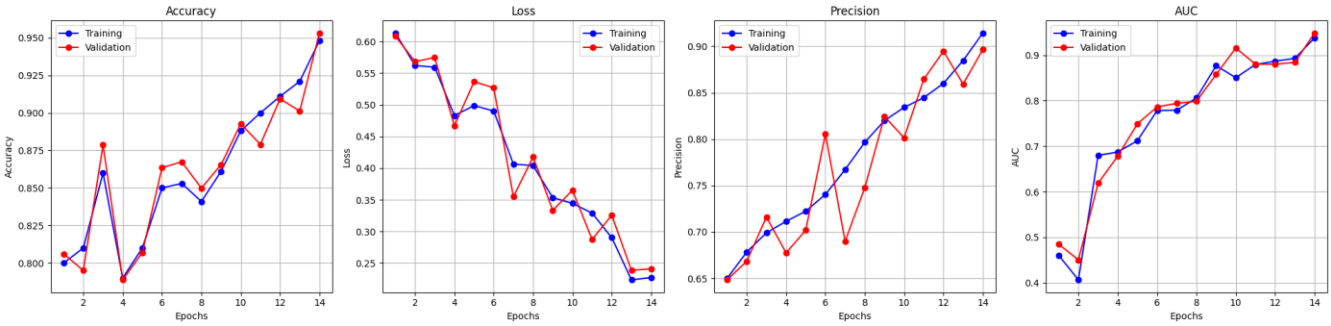
Graph 2.3: Model Accuracy Rate Change

As can be seen in the graph, the model accuracy starts to converge after the 25th iteration.

The model, which was trained for 3 epochs in each iteration and reached an accuracy rate of 0.6 after 37 iterations, increased the number of data approximately 2.8 times and increased the number of data from 2206 to 6177. In addition, the initial data numbers of 412, 318, 322, and 1154 were automatically organized as 1588, 1413, 1677, and 1499.

The artificial intelligence model, which was retrained with the automatically edited data set and trained with this data set for 14 ages, increased the accuracy rate of the manually edited data set from 92.1% to 94.8%.

The training process of the AI model trained with the new dataset using the previous parameters can be seen in the graph on the next page.



Graph 2.4: Training Process of DenseNet-169 with Edited Data Set

As can be seen from the graph, the algorithm that organized the data set increased the success rate.

2.8 Recommendations for the Future

The number of epochs can be increased by training the model for 3 epochs in each iteration. This can be realized with the help of computers with powerful hardware and automation performance can be increased.

Also, model performance can be increased by adding SMOTE and SMOTEEN techniques. Again, this is possible with powerful hardware features. An algorithm can be written to ensure a uniform distribution of the test data set and integrated into the general model. Because, if the classes in the test dataset are also irregular, the training process becomes inefficient.

In general artificial intelligence models, data augmentation is done automatically. In our model, the automatic data augmentation also depends on the number of data in the classes. And it takes place in the aforementioned while loop. There is no data incrementation [11].

In another scenario that might be appropriate to try, data augmentation can be performed both with the data augmentation code in the while loop and with the data augmentation method that already exists in the literature.

KAYNAKLAR

[1] Bergstra, J., Yamins, D., & Cox, D. D. (2013). Hyperopt: A python library for optimizing the hyperparameters of machine learning algorithms. In *Proceedings of the 12th Python in science conference* (Vol. 13, p. 20).

[2] Chawla, N. V., Bowyer, K. W., Hall, L. O., & Kegelmeyer, W. P. (2002). SMOTE: synthetic minority over-sampling technique. *Journal of artificial intelligence research*, *16*, 321-357.

[3] Wong, A., Kamel, M. S., Sun, Y., & Wong, A. K. C. (2011). Classification of imbalanced data: a review pattern-directed aligned pattern clustering view project pattern discovery in gene expression data view project classification of imbalanced data: a review. *International Journal of Pattern Recognition and Artificial Intelligence*, *23*(4).

[4] Pan, S. J., & Yang, Q. (2010). A survey on transfer learning. *IEEE Transactions on knowledge and data engineering*, *22*(10), 1345-1359.

[5] O'Shea, K., & Nash, R. (2015). An introduction to convolutional neural networks. arXiv preprint arXiv:1511.08458.

[6] He, H., Bai, Y., Garcia, E. A., & Li, S. (2008, June). ADASYN: Adaptive synthetic sampling approach for imbalanced learning. In *2008 IEEE international joint conference on neural networks (IEEE world congress on computational intelligence)* (pp. 1322-1328). IEEE.

[7] More, A. (2016). Survey of resampling techniques for improving classification performance in unbalanced datasets. *arXiv preprint arXiv:1608.06048*.

[8] Fernández, A., Garcia, S., Herrera, F., & Chawla, N. V. (2018). SMOTE for learning from imbalanced data: progress and challenges, marking the 15-year anniversary. *Journal of artificial intelligence research*, *61*, 863-905.

[9] Liashchynskyi, P., & Liashchynskyi, P. (2019). Grid search, random search, genetic algorithm: a big comparison for NAS. *arXiv preprint arXiv:1912.06059*.

[10] Polikar, R. (2012). Ensemble learning. In *Ensemble machine learning* (pp. 1-34). Springer, Boston, MA.

[11] Dao, T., Gu, A., Ratner, A., Smith, V., De Sa, C., & Ré, C. (2019). A kernel theory of modern data augmentation. In *International Conference on Machine Learning* (pp. 1528-1537). PMLR.