

# Contents

<b>1</b>	<b>Introduction</b>	<b>1</b>
<b>2</b>	<b>The dataset</b>	<b>1</b>
<b>3</b>	<b>Task 1: Related works</b>	<b>1</b>
<b>4</b>	<b>Task 2: Scratch CNN</b>	<b>1</b>
4.1	Data Preprocessing . . . . .	1
4.2	Task 2.1 . . . . .	1
4.3	Task 2.2 . . . . .	2
<b>5</b>	<b>Task 3: Pretrained network</b>	<b>2</b>
<b>6</b>	<b>Task 4: Baseline patches</b>	<b>2</b>
<b>7</b>	<b>Task 5: Ensemble network</b>	<b>2</b>

# 1 Introduction

# 2 The dataset

# 3 Task 1: Related works

# 4 Task 2: Scratch CNN

We developed different models of neural networks to work with the given dataset from scratch and we did different test to find the better hyperparameters to build the final model. This section is divided in three parts: in the first one we describe the main preprocessing applied to the data before the training of the models. Then we show how we built a model for classifying the dataset images between *mass* abnormality and *calcification* abnormality. In the last part, we describe the model built to classify between *benign* and *malignant* abnormality.

## 4.1 Data Preprocessing

Starting from the numpy arrays of the dataset images we were given, first of all we deleted all the samples associated with the *baseline patch* label. Since this labeled images were placed in the even positions of the dataset, we just selected all the odd-index samples and discard the others. This is done both for the training data and the test data.

Then we aggregated the labels according to the classification that we needed to do: in the *Task 2.1* the classes *mass benign* and *mass malignant* were aggregated in a unique class *mass*, and so for the *calcification* classes. On the other hand, in the *Task 2.2* we aggregated *mass benign* and *calcification benign* in the *benign* class, and the other labels in the *malignant* class.

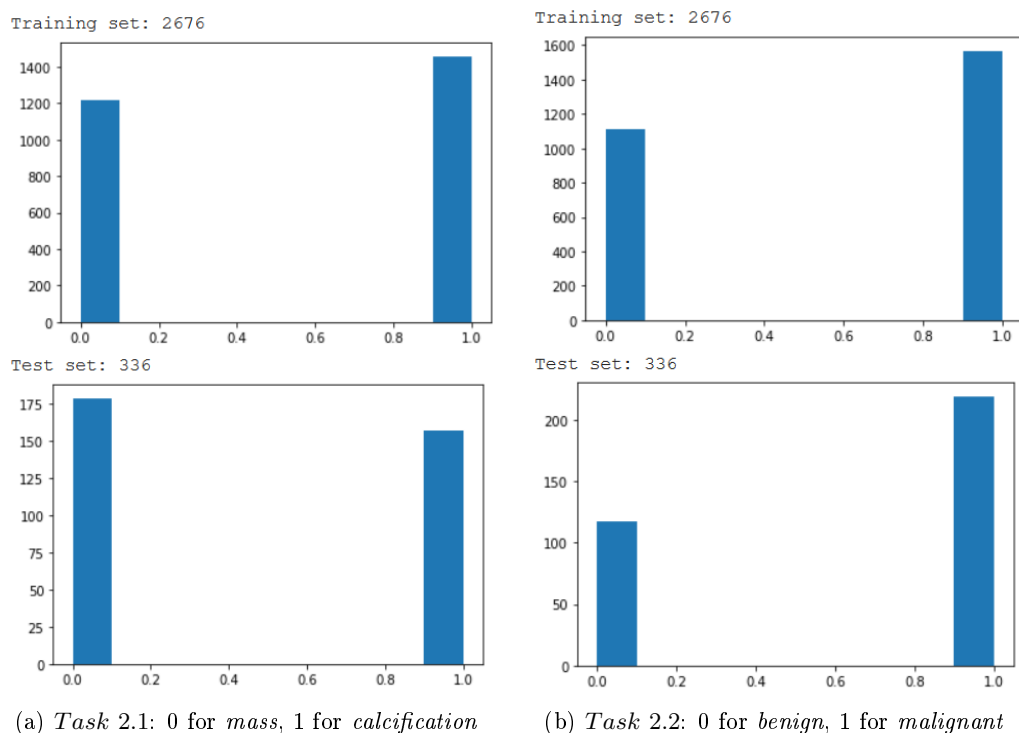


Figure 1: Label distribution

For the first case, we can see from the figure that the labels are pretty much equally distributed. Instead, in the second case the dataset is a little more unbalanced towards the *benign* class.

## 4.2 Task 2.1

For the first classification, we first tried to build a simple CNN

4.3 Task 2.2

5 Task 3: Pretrained network

6 Task 4: Baseline patches

7 Task 5: Ensemble network