CS550 Project Report – Turkish Lira Classification

Doğa Yılmaz - Cengiz Emre Dedeağaç

December 13, 2023

1 Introduction

Even after the pandemic came out, cash is still an essential part of transactions in today's daily life. People exchange various types of cash when shopping depending on the price and how much money they have in their pocket. It can be challenging for visually impaired people to distinguish easily and use the correct amount of cash trustfully during these transactions. To help people with these kinds of struggles, we want to create a tool where people can scan their banknotes and learn how much it is.

In this project, we prepared an application that satisfies the basic requirements for such service. For the purposes of this course, we utilized cloud services in our project. Our final product can recognize all Turkish banknotes of different values and differentiate a particular set of fakes.

2 Utilized Tools and Services

2.1 Amazon Rekognition

Amazon Rekognition is a service for training machine learning models for image and video processing. It accepts labeled images and trains a model to categorize any input image. It is effortless to use but might need some readjustments on the training data for better performance. This service acts as the core of our decision mechanism.

2.2 AWS S3

This is a cloud storage service provided by Amazon. S3 can be easily integrated with other services provided by Amazon. We use it to store training/test data.

2.3 Python

Python is one of the most popular programmings languages of recent years. It has a wide range of libraries and easy to type syntax for faster prototyping.

2.3.1 Django

Django is a very popular python framework that allows users easily create scalable websites. Which was a good choice for us because it is easy to test and host it to be used by other people.

3 Solution

In this section we will elaborate on the datasets we used, the AWS Rekognition Project, our Django client and the pricing of our solution.

3.1 Dataset

Throughout this project we created 3 different datasets by using several methods. In this section we will elaborate on each dataset in detail.

3.1.1 Dataset 1: Web Crawled Images Combined with Self Taken Images

Our initial approach was collecting Turkish Lira images from web and using them as our dataset. However, when we searched for high-quality images we could not find a diverse range of images. There are some images available but most of them are taken from the same angle or they have very low quality. Thus in order to make our dataset better we took images of each TL banknote and combined our images with web crawled images. The sample count for each class can be found in the table below.

Class Name	Sample Count
5 TL	28
10 TL	26
20 TL	23
50 TL	26
100 TL	32
200 TL	27

As it can be seen from the table above, we tried to pick sample count close to each other. Also for a traditional ML model sample counts are extremely low. However by using transfer learning AWS Rekognition does not require huge datasets in order to train a model which performs sufficiently.

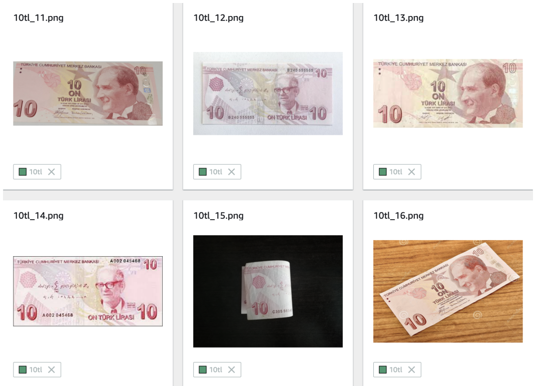


Figure 1: Sample images from dataset 1

3.1.2 Dataset 2: Self Taken Images

Similar to first dataset we have created an other dataset by only using the images taken by us. Our aim was to decrease the number of very similar data samples (can be seen in figure 1) and also to increase the image quality. Also we added an extra class for detecting fake banknotes which can be seen in figure 2. The additional class adds extra functionality to our model by supplying data to AWS Rekognition model for fake banknote instances. The sample count of for each class in dataset 2 can be found in the table below.

Class Name	Sample Count
5 TL	22
10 TL	19
20 TL	20
50 TL	20
100 TL	21
200 TL	25
Fake	37

3.1.3 Dataset 3: Self Taken Images with Bounding Box Data

AWS Rekognition Data Management tool has a feature which we can draw bounding boxes around the object we want to recognise. Thus we used this tool to draw and add bounding box information to our second dataset. One sample with a bounding box can be seen in the figure 4. By providing

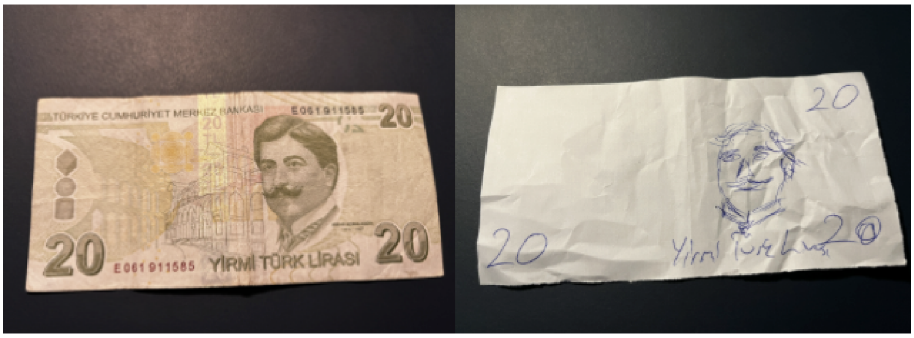


Figure 2: Original 20 TL banknote vs Fake 20 TL Banknote

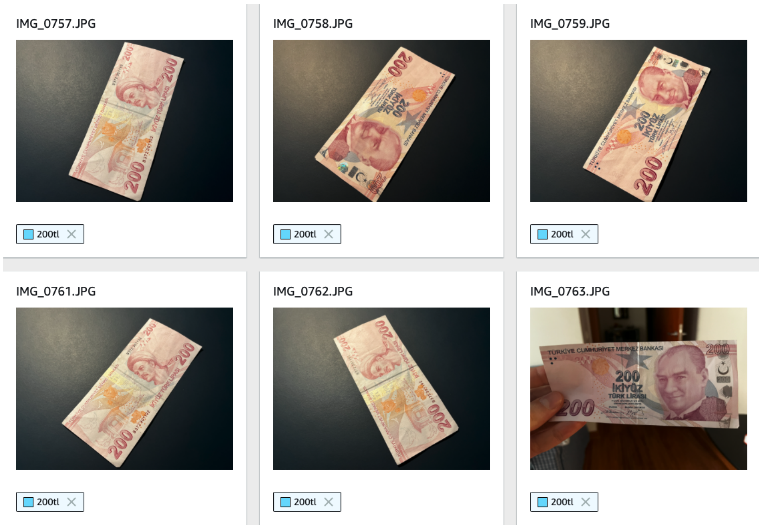


Figure 3: Sample images from dataset 2

extra information about the whereabouts of the banknote in the given image, we aim to improve the performance of our AWS Rekognition model in real world.

Figure 4: Image sample from dataset 3

3.1.4 Managing Datasets

Throughout this project, we have created a lot of experimental datasets. At some point dataset management become an issue. That is due to the AWS dataset management system. At the time we started this project in the mid semester, the system did not allow us to delete any dataset from the Web interface. Even though we delete the actual data in the s3 bucket, the dataset still stays visible in the AWS console. In order to solve this issue we headed to the AWS documentation however we could not find any solution there. Following that we have posted a question to the StackOverflow regarding this issue.

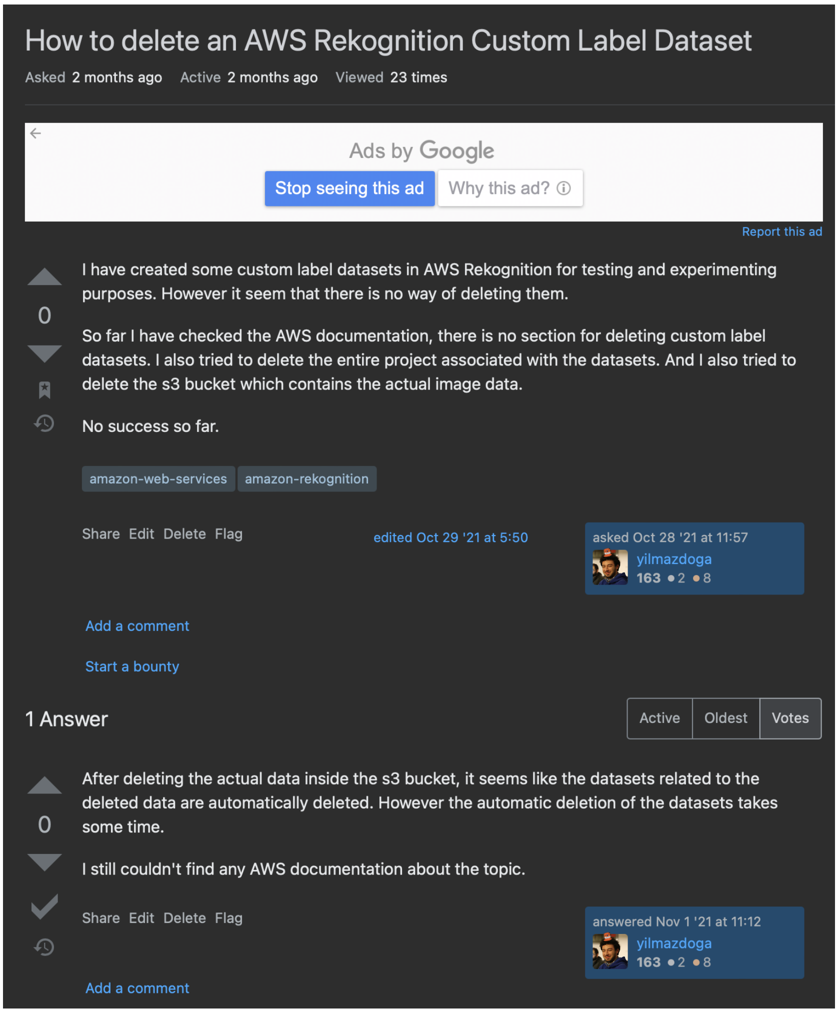


Figure 5: Our StackOverflow question.

Around late semester an update is made by AWS which changed the management of the datasets. Currently datasets are managed under AWS Rekognition projects and they can be deleted or modified from the AWS console. We think that this is a nice example that how the cloud products change according to the needs of the users and we thought that the situation is worth reporting.

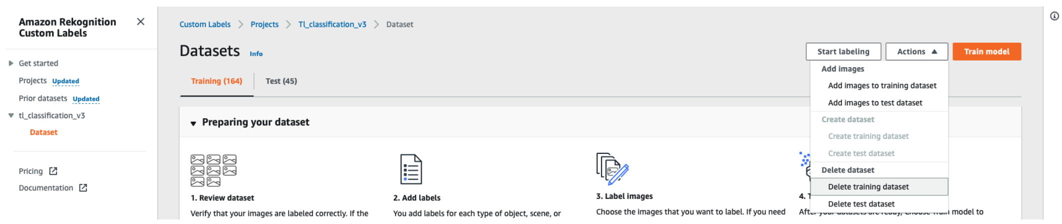
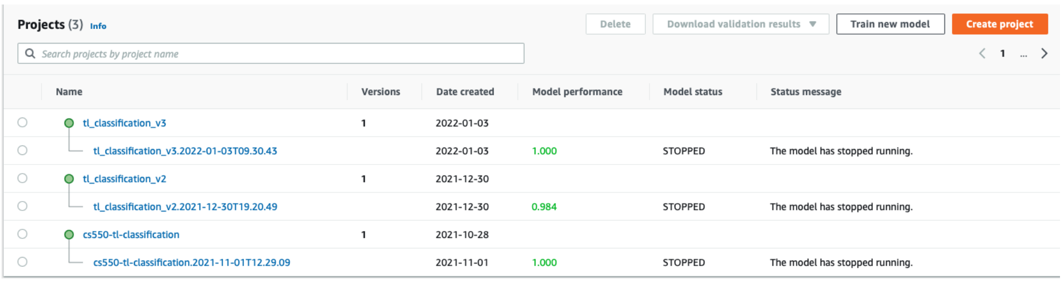


Figure 6: Updated AWS Rekognition console

3.2 AWS Rekognition Project and Training Progress

Throughout this project we have trained 3 different model with our 3 datasets. The training process takes around 1 hour. Our 3 models can bee seen in the figure 7.

With AWS Rekognition we can train sophisticated models with a push of a button. The system uses transfer learning in order to train models with very limited number of data.


Projects (3) Info

Delete | Download validation results | Train new model | Create project

Search projects by project name

Name	Versions	Date created	Model performance	Model status	Status message
tl_classification_v3	1	2022-01-03			
tl_classification_v3.2022-01-03T09.30.43		2022-01-03	1.000	STOPPED	The model has stopped running.
tl_classification_v2	1	2021-12-30			
tl_classification_v2.2021-12-30T19.20.49		2021-12-30	0.984	STOPPED	The model has stopped running.
cs550-tl-classification	1	2021-10-28			
cs550-tl-classification.2021-11-01T12.29.09		2021-11-01	1.000	STOPPED	The model has stopped running.

Figure 7: AWS Rekognition console screenshot of our models

3.3 Django Client

In order to use our trained models, we need a software which uses a camera to take the image of a TL banknote and sends the captured image to the AWS Rekognition model. To achieve this functionality we used Django library and made a very simple web app which opens up the webcam of the system and captures images every 2 seconds. Than we used Boto3 (Python SDK for AWS) in order to send our captured images to our AWS Rekognition model. Once the response arrives to our client we store it in a local variable. If two consecutive frames are classified with same label the client shows the class label at top left of the screen. Also if the Rekognition service is offline the client shows a warning to the user. A screenshot of our Django Client can be found in figure 8. Also in order to show the real world performance, we prepared a demo video which we use the Django client with our AWS Rekognition model. The demo video is accessible via the following link: https://youtu.be/JPPh7ovYRg4. Also our Django Client implementation is accessible via the GitHub link: INSERT GitHub LINK HERE. TODO.

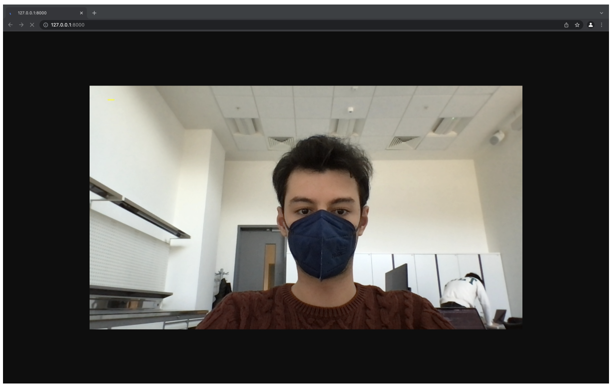


Figure 8: Screenshot of our Django Client

3.4 Pricing of Our Solution

When using cloud services, pricing is also an important matter. in this section we will provide information about the prices of each building block of our solution. The unit prices of our building blocks are listed below.

Service Name	Price
AWS S3 Storage	\$0.023/GB
AWS S3 Data Read	\$0.0004/1,000 request
AWS S3 Data Write	\$0.005/1,000 request
AWS Rekognition Custom Label Training	\$1.00/hr
AWS Rekognition Custom Label Inference	\$4.00/hr

As it can be seen from the table above, data storage, read and write costs are manageable even if we have a very large dataset which we do not need with AWS Rekognition's transfer learning feature. However the Rekognition service itself is an expensive one. It costs 1 USD per hour during the model training and 4 USD per hour while our prediction instance is up. However, AWS Free tier provides a limited amount of free of charge usage each month. Free tier provides 5GB of storage, 20000 read and 2000 write requests for S3 service. It also provides 10 hours of training time per month and 4 hours of inference time per month for the first 3 months. By using the AWS Free tier we managed to stay at zero cost throughout the semester.

4 Results

We were able to connect to the Rekognition service and detect different banknotes and fake versions using our trained model.

5 Conclusion and Future Work

References