MEF UNIVERSITY

BOARDING PASS DETECTION IN SOCIAL MEDIA TO PREVENT FLIGHT INFORMATION THEFT

Capstone Project

Hasan Oktay Ekici

İSTANBUL, 2019

MEF UNIVERSITY

BOARDING PASS DETECTION IN SOCIAL MEDIA TO PREVENT FLIGHT INFORMATION THEFT

Capstone Project

Hasan Oktay Ekici

Advisor: Asst. Prof. Dr. Tuna Çakar

İSTANBUL, 2019

MEF UNIVERSITY

Name of the project: Boarding Pass Detection In Social Media To Prevent Flight Information Theft
Name/Last Name of the Student: Hasan Oktay Ekici
Date of Thesis Defense: 09/09/2019

I hereby state that the graduation project prepared by Hasan Oktay Ekici has been completed under my supervision. I accept this work as a "Graduation Project".

__/__/____
Asst. Prof. Dr. Tuna Çakar

I hereby state that I have examined this graduation project by Hasan Oktay Ekici which is accepted by his supervisor. This work is acceptable as a graduation project and the student is eligible to take the graduation project examination.

__/__/____

Director
of
Big Data Analytics Program

We hereby state that we have held the graduation examination of Hasan Oktay Ekici and agree that the student has satisfied all requirements.

THE EXAMINATION COMMITTEE

Committee Member	Signature
1. Asst. Prof. Dr. Tuna Çakar	……..
2. Prof. Özgür Özlük	……..

Academic Honesty Pledge

I promise not to collaborate with anyone, not to seek or accept any outside help, and not to give any help to others.

I understand that all resources in print or on the web must be explicitly cited.

In keeping with MEF University's ideals, I pledge that this work is my own and that I have neither given nor received inappropriate assistance in preparing it.

Hasan Oktay Ekici 09/09/2019

Name Date Signature

EXECUTIVE SUMMARY

BOARDING PASS DETECTION IN SOCIAL MEDIA TO PREVENT FLIGHT INFORMATION THEFT

Hasan Oktay Ekici

Advisor: Asst. Prof. Dr. Tuna Çakar

SEPTEMBER, 2019, 25 Pages

During the past few years, along with social media gaining bigger share on people's lives, everyone has started to share their moments with detailed information on multiple platforms instantly. Sharing these kinds of information on posts may cause security bugs in people's lives, such as undesired flight changes/cancellations because of flight information theft, frequent flyers miles theft, and even car theft and burglary. This project's aim is to develop an artificial intelligence algorithm that can help to prevent these security bugs.

In this project, we use data that is collected from instagram posts that contain boarding passes. Our main purpose is to build an artificial intelligence that makes decisions and processes following procedures: a machine learning algorithm that decides if the shared instagram post contains a boarding pass shared with #boardingpass; an optical character recognition algorithm that gathers text information from the post and scripts that send the information instantly to the relevant air carrier about the shared post. With this information, air carrier will be able to inform the passenger about their concern on the flight safety only in a couple of minutes after the post is shared.

Key Words: Object Detection, Boarding Pass Detection, Flight Information Theft, Machine Learning, Neural Networks, Tensorflow, Artificial Intelligence.

ÖZET

SOSYAL MEDYADA UÇUŞ KARTI (BİNİŞ KARTI) SAPTANMASI İLE UÇUŞ BİLGİLERI HIRSIZLIĞININ ÖNLENMESİ

Hasan Oktay Ekici

Tez Danışmanı: Dr. Öğr. Üyesi Tuna Çakar

EYLÜL, 2019, 25 Sayfa

Son yıllarda sosyal medyanın hayatımızın tüm aşamalarında büyük pay sahibi olması ile insanlar her anlarını bir çok platformda gerçek zamanlı olarak ve tüm bilgileriyle paylaşmaya başladı. Bu tarz paylaşımların insanların hayatlarında güvenlik açıkları oluşturabildikleri bilinmektedir. Bu çalışmanın amacı Instagram gibi popular bir fotoğraf, an ve video paylaşma platformunda uçuş kartlarının, uçuş gerçekleşmeden önce paylaşılmasının, uçuş bilgilerinin üçüncü kişilerce ele geçirilmesi sonucu uçuş iptali, uçuş değişiklikleri, uçuş millerinin çalınması, uçuş tarihlerinin bilinmesi vesilesiyle ev, araç hırsızlığı gibi istenmeyen olayların önüne geçmeye yardımcı olacak bir yapay zeka algoritması oluşturmaktır.

Çalışmada instagramda #boardingpass bilgisi ile paylaşılan fotoğraflarda uçuş kartlarının olup olmadığının makine öğrenmesi ile tespit edilmesi, uçuş kartı tespit edilirse optik karakter tanıma ile uçuş kartının içindeki bilgilerin alınması ve ardından ilgili havayoluna raporlanması işlemlerine karar verecek ve yürütecek bir yapay zeka tasarlanması hedeflenmektedir. Bu sayede havayolu da yolcuya ilgili fotoğrafın paylaşılmasının ardından birkaç dakika içerisinde uçuş güvenliği açısından bilgilendirme yapabilecektir.

Anahtar Kelimeler: Nesne Algılama, Uçuş Kartı Algılama, Uçuş Bilgileri Hırsızlığı, Makina Öğrenmesi, Yapay Sinir Ağları, Yapay Zeka.

TABLE OF CONTENTS

LIST OF TABLES AND FIGURES

Tables

Figures

1. INTRODUCTION

Social media has become an integral part of people's lives all over the world. With the establishment of the social media platform "Six Degrees" in 1997, the social media on www has started its life [1]. Today social media usage has reached enormous numbers. Social media users are expected to be more than 3 billion in 2021 [2]. The number of active Instagram users in a month has passed 1 billion limit in June 2018 [3] and 500 million of these users are active daily. Since more than 100 million posts are shared and on the average 53 minutes per user are spent daily, lots of different kinds of information can be reached via these posts.

Recent years many incidents showed us that sharing important information on social media may end up some undesired incidents [4]. While travelers share moments of their holidays, they also share their passports with boarding passes at the airports. When we conduct a search on Instagram with #boardingpass hashtag, we can see that more than 110.000 photos are posted as of June 2019. In these posts, users share their boarding passes that contain barcode, PNR (Passenger Name Record), name, date and other flight information, which are necessary in order to change, modify or even cancel user's flight by third party. Recently Kaspersky published an article that shows sharing boarding passes on social media may lead to some security issues such as:

- It may give insight to burglars or car thieves about your absence in town.
- Your seats or flight dates can be changed using your flights information.
- Wild-cat cancellation of your flights.
- Your frequent flyer miles can be transferred using your information [5].

1.1. Literature Survey

Although, to the best of our knowledge, the literature does not include any other study related to the one studied in this project, steps of the project are held separately in various studies.

1.1.1. Object Detection

Object detection is a computer vision technique to identify pre-defined objects in an image or a video. While the classical object detection systems search for certain scales and

aspect ratios [6][7], the Convolutional Neural Networks (CNN) showed a higher performance on multiclass and multiscale problems. [8].

The main obstacle in object detection is that object can appear in different sizes and shapes in images. There are two solutions for detecting the object, first way is to rescale image for multiple scales to fit or to apply classifier in different shapes to fit to image [8].

1.1.2. Optical Character Recognition

Optical Character Recognition (OCR) systems extract text characters from images and videos. OCR systems are used widely in technology such as identifying number plates in traffic cameras, gathering passport information on borders, simultaneously translation applications and id card detection. But gathering text from OCR systems have some common problems when the text size or text font varies, the text is skewed and there are other objects in the images [9].

If the image contains objects other than text, binarization with thresholding can reduce the visibility of the objects with low contrast, however, binarization increases the contrast between text and the paper and becomes more visible [10].

2. PROJECT STATEMENT

In this section, the objective and the scope of the project are discussed.

2.1. Project Objective

The objective of the project is to develop an Artificial Intelligence algorithm to detect posts on Instagram that contain boarding passes and send the boarding pass information to the air carriers and passengers to prevent flight information theft within a couple of minutes after the posts are shared.

The steps of the algorithm are as follows:

- Downloading the Instagram posts that have #boardingpass hashtag
- Conducting Object Detection algorithm to classify if the post indeed has a boarding pass
- Saving only the detected boarding pass as a new image
- Binarization and deskewing of the new image
- Conduct OCR to convert image into text
- Filtering Keywords
- Sending gathered information to the related air carrier

2.2. Project Scope

The scope of this study covers developing an artificial intelligence algorithm that can detect the boarding passes and extract the text inside the boarding pass. A major difficulty for detecting a boarding pass is that existing libraries cannot detect boarding passes as an object in photographs. An object detection algorithm determines whether the image contains a boarding pass by the medium of an image recognition library that is trained exclusively for this study. After detection of the boarding pass, the algorithm cuts out the boarding pass from the image and saves as a new image in order to focus on only the boarding pass rather than the full image.

The new image gets binarized with a specified threshold to concentrate on the text ground. A deskewing algorithm detects the slope of the text and deskews the slope to horizontal axis. The OCR algorithm extracts text in the binarized and deskewed image. After the text is recognized in the image, a python script decides if the recognized text contains

specified characters or terms and proceeds to send the relevant information with the stated air carrier.

2.3. Instagram

Instagram is a social media application for sharing photos and videos launched in 2010. It allows users to upload photos, videos or stories which can be edited with filters and share geo-locations and tags (hashtag)[11]. Users can freely search within the posts via hashtags, usernames and locations. The hashtag is used to categorize the content of the posts.

2.4. Dataset

The dataset consists of 6.026 images, which are downloaded from Instagram via instaloader script. The "#boardingpass" and "#boardingpasses" hashtags were used to find relevant posts with high probability of containing boarding passes. The images are dated with a wide period from 21/06/2012 to 22/06/2019.

After a careful selection of images, the dataset is split into "containing boarding pass" and "not containing boarding pass" sets. Before proceeding with further analysis, we provide the details of the dataset below:

- 3250 images with #boardingpass hashtag that do not contain boarding pass
- 409 images with #boardingpass hashtag that contain boarding pass
- 1664 images with #boardingpasses hashtag that do not contain boarding pass
- 703 images with #boardingpasses hashtag that contain boarding pass

The dataset that includes the images containing boarding passes is split into test and train datasets randomly. The test dataset consists of 248 images and xml's while the train dataset consists of 845 images and xml's.

While collecting images, instaloader also downloads images that are shared with French and Spanish boarding pass hashtag "#cartedembarquement" and "#pasedeabordar" respectively. These images have not been taken into consideration at first stage due to different linguistic structure; however, they are kept as backup for model training needs.

2.5 Exploratory Data Analysis

We need to understand the image distribution in the dataset in order to apply suitable machine learning algorithms for training the model. The quantities of images with #boardingpass and #boardingpasses are as follows:

6.026 images are downloaded during the data collection stage.

4.914 images do not contain boarding passes.

1.112 images contain boarding passes.

Moreover, images with "#cartedembarquement" and "#pasedeabordar" are downloaded in numbers specified below:

66 images contain boarding passes and are tagged as "#cartedembarquement"

432 images do not contain boarding passes and are tagged as "#cartedembarquement"

73 images contain boarding passes and are tagged as "#pasedeabordar"

1.185 images do not contain boarding passes and are tagged as "#pasedeabordar"

For further understanding the mood while sharing the posts, we conducted analyses on hashtags shared with the post. As seen in Figure 1 most shared hashtags are #boardingpass and #boarding passes along with #travel, #passport and #airport. The figure tells us boarding passes are shared mostly while people are travelling or when they are in travelling mood.

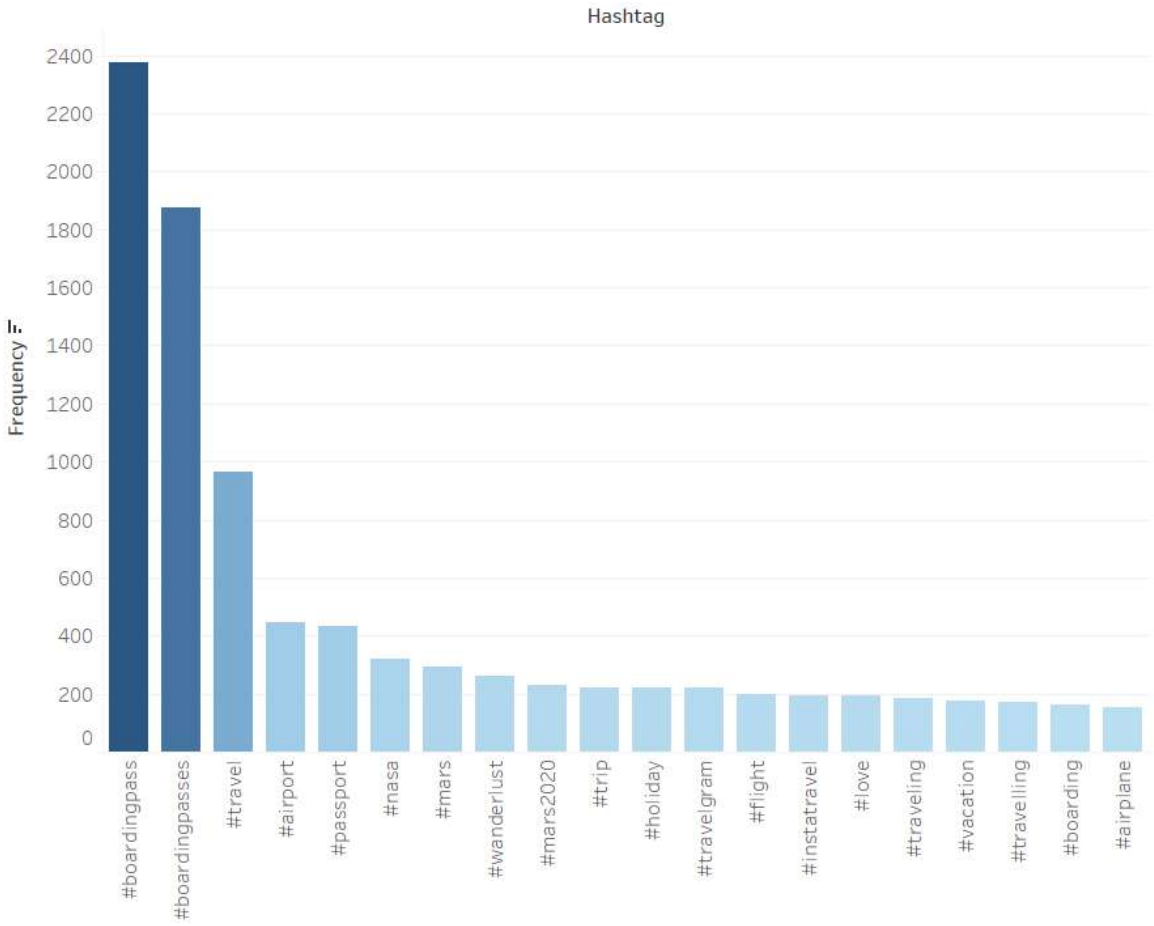

Figure 1 Most recurring hashtags shared within the posts with images that are posted with #boardingpass and #boardingpasses

Moreover, an analysis on the location keywords that the posts were shared shows us that users mostly share their boarding passes at airports as expected (see Figure 2).

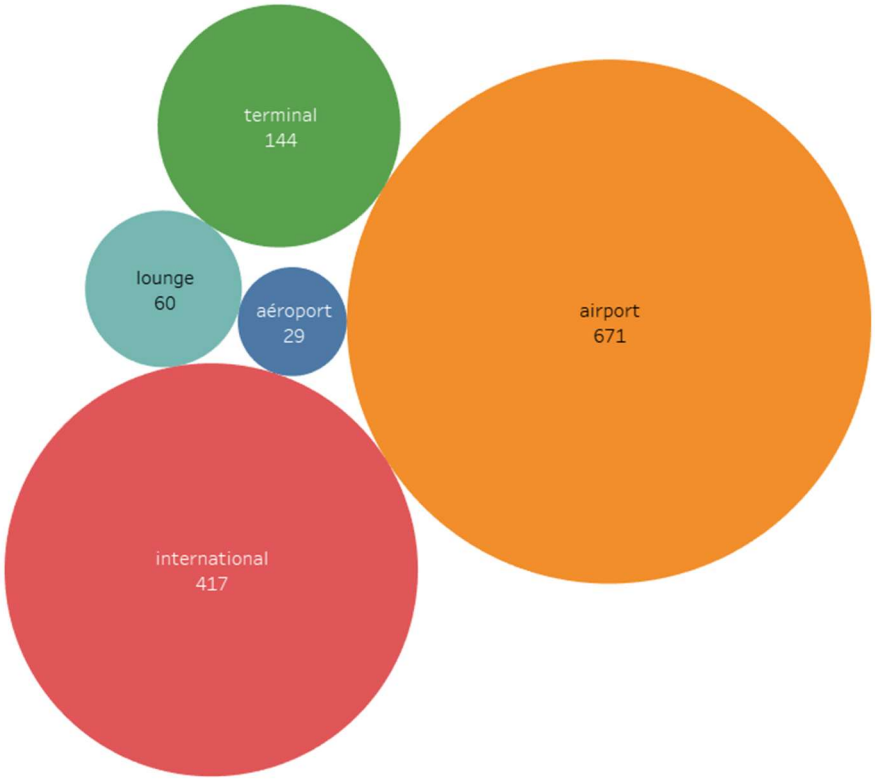

Figure 2 The top 5 most reccuring locations (in terms of types building types)

When we conduct further analysis on locations in terms of geographical locations, we see that Italy and London share the first place as most post shared locations (see Figure 3). Bangkok, California, New York and Spain are following the top tiers as most boarding pass shared locations.

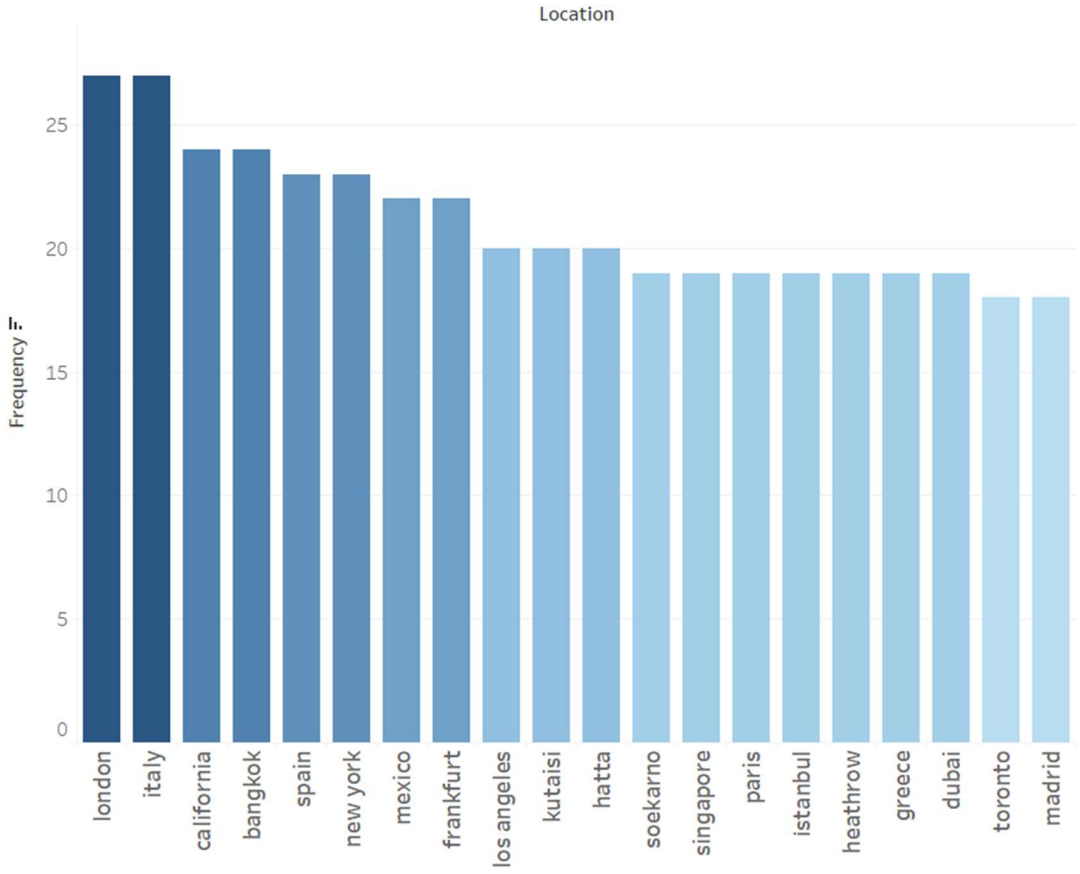

Figure 3 Most #boardingpass shared locations (in terms of geographical locations)

3. METHODOLOGY AND APPLICATION

A six-stage approach with several machine learning algorithms and models are used in the project as well as python scripts in order to gather data, process data, train model, execute classifiers and forward the processed data to air carriers.

3.1. Methods and Techniques

The Artificial Intelligence in this project consists of six stages of processes (see Figure 4). During these stages; (i) an Instagram post downloading script "instaloader" is used for downloading the posts, (ii) a Neural Network Machine Learning Algorithm (Faster R-CNN algorithm that we train) is used for object detection if a downloaded image contains a boarding pass and save the detected boarding pass as a new image (iii) a script to binarize and deskew the image (iv) an Optical Character Recognition Algorithm (OCR along with pytesseract library) is used for text mining and revealing the text in documents, (v) an algorithm if the mined text contains air carrier name, ticket number and/or PNR, finally (vi) a script to send the gathered data to the air carrier so that the air carrier may contact and inform the passenger.

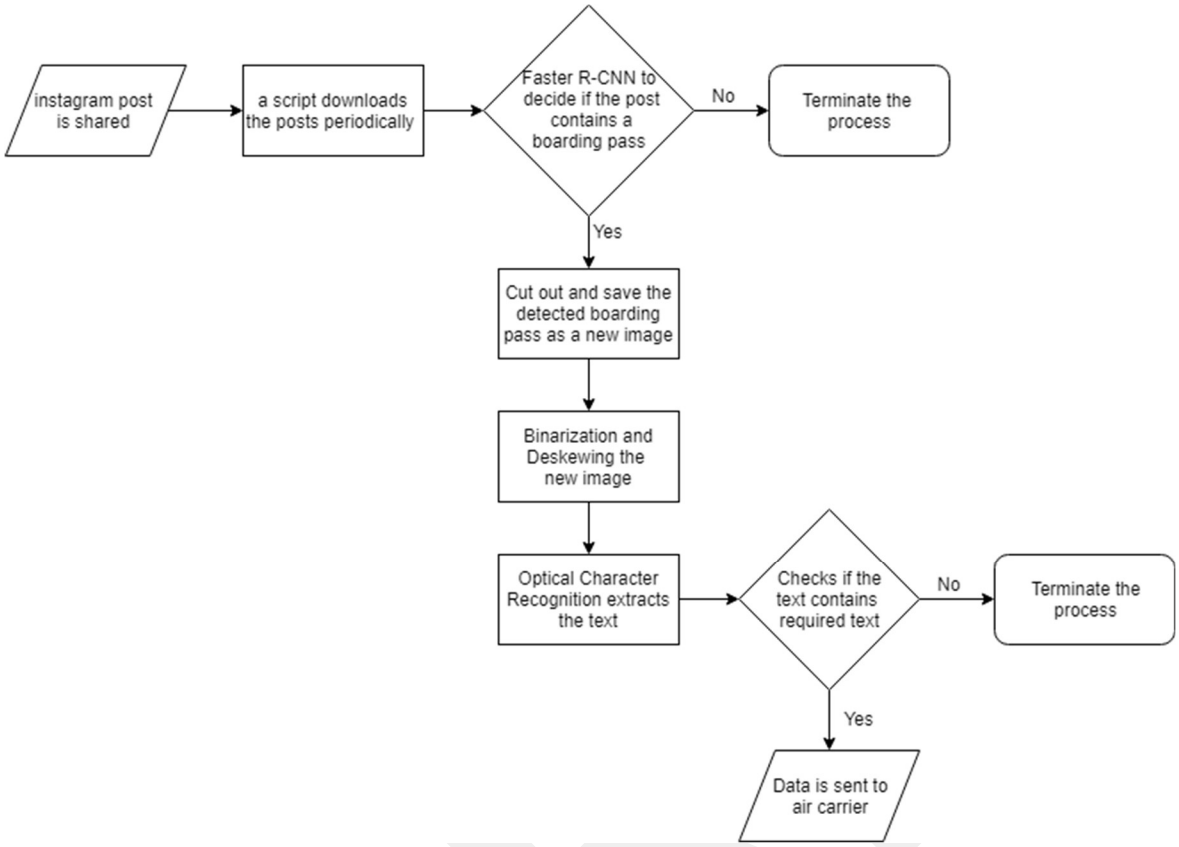

Figure 4 The main processing flow of the Artificial Intelligence

The whole algorithm is planned to conduct all processes within two minutes; thus, the passenger is informed as rapidly as possible after sharing the boarding pass on Instagram to reduce the risk of flight information theft.

3.1.1 Instaloader

Instaloader is a python package written by Alexander Graf (aandergr) to download desired Instagram posts with various parameters. We used basic downloading package, which downloads the newest posts shared with #boardingpass. We use the package both for acquiring the posts to create the dataset and to download the post to run the AI.

While acquiring the posts for dataset, the algorithm is allowed to download a limited number of posts due to Instagram restrictions. Therefore, we run instaloader once every week to collect fresh posts.

3.1.2 Faster R-CNN (Tensorflow)

We use a Convolutional Neural Network (CNN) to conduct Object Detection as a classification algorithm in the Instagram posts. CNN is a machine learning algorithm in Deep Learning category. There are various CNN algorithms that are able to detect various objects such as R-CNN, Fast R-CNN, Faster R-CNN, SSD, YOLO and RetinaNet. The algorithms' performances vary on the type of input, CPU, GPU, script language and many more factors.

While CNN is a high performing machine learning algorithm for image recognition, R-CNN focusses on regions and acts better for detecting objects. R-CNN searches the image in rectangular shapes and detects the objects. Faster R-CNN requires a lower training time and performs faster than the other algorithms [12].

In this project Tensorflow with Faster R-CNN on Resnet 101 Coco model is used in order to detect the images fast and with high accuracy.

Although we run some of the pre-trained models for object detection such as Faster R-CNN and SSD, they are not be able to detect boarding passes therefore we need our personalized model. We train the model with our own dataset so that our model would be able to detect boarding passes with very high accuracy. Our training dataset contains 845 images and 845 xml files while our test dataset contains 248 images and 248 xml files. We downloaded the images via instaloader script and split into test and train sets randomly. With the necessity of labeling the images, 1094 images are labeled by hand and xml files were generated through an image labelling application called "LabelImg" manually.

We trained the model in a powerful computer with intel i9 processor and Nvidia GTX1080 GPU. Although the loss in the model is decreased to less than 0.1 in 25.000 steps, we continued to achieve 56.000 steps to strengthen the trained model (see Figure 5)

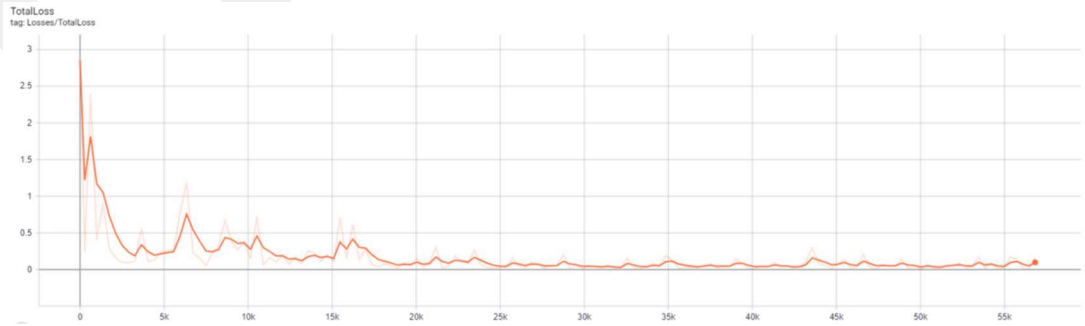

Figure 5 Training loss graph

After the model is trained, the algorithm is able to detect the boarding pass patterns as an object. As we can see in Figure 6 that the model is able to detect the boarding passes with a very high accuracy.

Figure 6 Detection of boarding pass with trained Faster R-CNN model

After the image recognition process, we want the model to focus on the boarding pass. Any object rather than a plain paper with text can distract the optical character recognition engine. Therefore, the script cuts out the detection box as a new image as seen in Figure 7.

Figure 7 Detected boarding pass that is saved as a new image

3.1.3 Binarization and Deskewing

While detected if a boarding pass has all the required information, it is not always easy for OCR to recognize the letters. For increasing the text recognition accuracy, we apply two separate processes that are Binarization and Deskewing.

Binarization is a Dimensionality Reduction technique to convert image into black and white while specifying a threshold. After binarization, only the high contrast objects, such as text, remain in the image. We applied this technique as seen in Figure 8 to increase the text recognition performance.

Figure 8 The binarized image

Although the image has been binarized, the text recognition performance can still be poor as the text in the image may be skewed. Deskewing technique detects the angle of the text in the image and rotates the image until the text becomes horizontally flat. This procedure lets the OCR to conduct with a higher performance and increases the text

recognition accuracy. Figure 9 presents a deskewed image which is ready for text recognition and extraction processes.

Figure 9 The deskewed image

3.1.4 Text Extraction and Further Actions

The text inside the image is text-mined via OCR. By using the OpenCV library and Tesseract (pytesseract script), we extract all flight information text from the boarding pass image.

The text, which is gathered from the boarding pass, is processed via a python script. The script searches for a specific word combination in the gathered text and takes necessary actions upon finding them.

In our study, we search for numbers starting with “235”, which specifies that the ticket number is related to Turkish Airlines. If the searched text is found, the script forwards all the information to the related air carrier with an e-mail.

After the OCR runs, the text is revealed, which can be seen in Figure 10. As the recognized text contains name, surname, ticket number and other relevant flight information, this information is enough to cancel flight or change connected flights. The algorithm will continue to send the text information to relevant air carrier for prevention of flight information theft.

```
Out[63]: 'OARDING PASS | BINIS KARTT       EKICI OKTAY APIS OK EKI 6 Y
18APR 12:35 SAW = Tk 731 GNY ve  | 11:50 ate SAW see “ 3 | |  sae ET
23523793277221 gan c'
```

Figure 10 The extracted text that is recognized by OCR

3.1.5 Model Consistency

For testing the model, we chose Prediction Status and Detection Accuracy. We run the model with ten randomly selected Instagram posts. In all of the post, the model correctly predicts whether there are boarding passes in the post or not. The Decision Matrix in Table 1 shows us the correctly predicted decisions. The Detection Accuracy shows the probability if the object is a boarding pass.

The average accuracy is over 99,9% while the precision at prediction is %100. The consistency in the model's detection accuracy and decision matrix is an indicator that model is ready to deploy and use.

Table 1 Results of the 10 randomly selected posts

Index	Does the post contain boarding pass?	Does the model find Boarding Pass?	Detection Accuracy	Decision Matrix
1	No	No	-	True Negative
2	Yes	Yes	99,999%	True Positive
3	Yes	Yes	99,999%	True Positive
4	No	No	-	True Negative
5	Yes	Yes	99,996%	True Positive
6	No	No	-	True Negative
7	No	No	-	True Negative
8	Yes	Yes	99,999%	True Positive
9	Yes	Yes	99,999%	True Positive
10	No	No	-	True Negative

4. CONCLUSION

In this study we build an artificial intelligence algorithm to detect the boarding passes shared on Instagram with #boardingpass and #boardingpasses hastags and extract flight information to prevent flight information theft. The Faster R-CNN model can detect the boarding pass with more than %99 accuracy.

The mostly encountered issue is that there are other objects shared on Instagram such as wedding invitation, concert or festival tickets which are not boarding passes but contain #boardingpass tags. This problem can be eliminated by determining the most used tags in these posts and avoiding downloading and processing these images.

Another obstacle for this model is skewed, blurred, distant and unclear images for extracting text. The OCR requires clear text without distortion and other objects for high performed text recognition processes.

Although the text recognition performance should be improved, the model can be personalized for any air carrier, is capable of detecting boarding passes and extracting information in less than a minute and ready to support air carriers to prevent flight information theft.

Additionally, a barcode scanner can extract the information from boarding pass and can improve the output quality of the algorithm.

REFERENCES

[1]https://www.socialmediatoday.com/news/the-history-of-social-media-infographic-1/522285/

[2]https://www.statista.com/statistics/278414/number-of-worldwide-social-network-users/

[3]https://www.statista.com/statistics/253577/number-of-monthly-active-instagram-users/

[4]https://www.forbes.com/sites/alexandratalty/2017/04/30/dangerous-airline-boarding-pass-hacking-trend-puts-travelers-at-risk/#52a905f75d2c

[5]https://www.kaspersky.com/blog/dont-post-boarding-pass-online/10495/

[6] Viola, P.A., Jones, M.J.: Robust real-time face detection. Int. J. Comput. Vis. 57(2), 137–154 (2004)

[7] Doll´ar, P., Appel, R., Belongie, S.J., Perona, P.: Fast feature pyramids for object detection. IEEE Trans. Pattern Anal. Mach. Intell. 36(8), 1532–1545 (2014)

[8] Cai Z., Fan Q., Feris R.S., Vasconcelos N. (2016) A Unified Multi-scale Deep Convolutional Neural Network for Fast Object Detection. In: Leibe B., Matas J., Sebe N., Welling M. (eds) Computer Vision – ECCV 2016. ECCV 2016. Lecture Notes in Computer Science, vol 9908. Springer, Cham

[9] Impedovo, S., Ottaviano, L., & Occhinegro, S. (1991). Optical Character Recognition - a Survey. *IJPRAI, 5*, 1-24.

[10] White, James M., and Gene D. Rohrer. "Image thresholding for optical character recognition and other applications requiring character image extraction." IBM Journal of research and development 27.4 (1983): 400-411.

[11]https://en.wikipedia.org/wiki/Instagram

[12] Ren, Shaoqing, et al. "Faster r-cnn: Towards real-time object detection with region proposal networks." Advances in neural information processing systems. 2015.

APPENDIX

Python code for Downloading posts:

```
import instaloader
L = instaloader.Instaloader()
L.interactive_login("develoktay")                    # asks password
for post in L.get_hashtag_posts("boardingpass"):     # hashtag to download
    L.download_post(post, target="#boardingpass")    # download directory
```

Python code for concatenation of txt files:

```
import glob
import shutil
import os
os.chdir("C:\\Users\\oktay\\iCloudDrive\\BDA\\Graduation Project\\Transferred images\\Capstone\\Hashtag analysis")   # specify directory
outfilename = 'all_txt_combined' + ".txt"            # file name
filenames = glob.glob('*.txt')

with open(outfilename, 'wb') as outfile:
    for filename in glob.glob('*.txt'):
        if filename == outfilename:
            continue
        with open(filename, 'rb') as readfile:
            shutil.copyfileobj(readfile, outfile)
```

Python code for hashtag analysis:

```
import pandas as pd
import os
import matplotlib.pyplot as plt

file = open('all_txt_combined.txt', encoding="utf8")     #open combined text file
Text = file.read()
```

```
for char in '-.,\n':
    Text=Text.replace(char,' ')
Text = Text.lower()                          # clean the words

word_list = Text.split()
textag = []                                  # create list to keep only words
for i in word_list:                          # start with hashtags
    if i[0] == "#":
        textag.append(i)

d = {}
for word in textag:
    d[word] = d.get(word, 0) + 1             # count words frequency
word_freq = []
for key, value in d.items():
    word_freq.append((value, key))

word_freq.sort(reverse=True)                 # sort words in descending frequency
print(word_freq)
wordcountdf = pd.DataFrame(word_freq)        # convert dict to dataframe
wordhead = wordcountdf.head(10)              # first 10 most used hashtags10
wordhead.columns = ['count','hashtag']       # rename column names
wordhead.plot(kind="bar")
```

Python code for boarding pass detection, binarization, deskewing, text detection and sending the text:

```
import os
import numpy as np
import tensorflow as tf
from PIL import Image
import cv2
from utils import label_map_util
```

```
from utils import visualization_utils as vis_util
import pytesseract
from pathlib import Path


modelname = 'inference_graph'                    # Our own trained model path
workdir = os.getcwd()

ckptpath = os.path.join(workdir,modelname,'frozen_inference_graph.pb')
                                                 # Path to our own trained model
labelpath = os.path.join(workdir,'training','labelmap.pbtxt')
                                                 # Path for labels


numclasses = 1                                   # Number of classes, 1 for boarding pass

label_map = label_map_util.load_labelmap(labelpath)
categories = label_map_util.convert_label_map_to_categories(label_map,
max_num_classes=numclasses, use_display_name=True)
category_index = label_map_util.create_category_index(categories)

detection_graph = tf.Graph()
with detection_graph.as_default():
  od_graph_def = tf.GraphDef()
  with tf.gfile.GFile(ckptpath, 'rb') as fid:
    serialized_graph = fid.read()
    od_graph_def.ParseFromString(serialized_graph)
    tf.import_graph_def(od_graph_def, name='')

  sess = tf.Session(graph=detection_graph)

image_tensor = detection_graph.get_tensor_by_name('image_tensor:0')
detection_boxes = detection_graph.get_tensor_by_name('detection_boxes:0')
```

```
        detection_scores = detection_graph.get_tensor_by_name('detection_scores:0')
        detection_classes = detection_graph.get_tensor_by_name('detection_classes:0')
        num_detections = detection_graph.get_tensor_by_name('num_detections:0')

        # ACTION LOOP
        pathlist = Path("C:\\tensorflow1\\models\\research\\object_detection\\#boardingpass").glob('**/*.jpg')
        for path in pathlist:
            # BOARDING PASS DETECTION
            image = cv2.imread(str(path))                    # Load image
            image_expanded = np.expand_dims(image, axis=0)   # Expand image

            (boxes, scores, classes, num) = sess.run(
                [detection_boxes, detection_scores, detection_classes, num_detections],
                feed_dict={image_tensor: image_expanded})
                                                 # Create boxes, scores, classes, num

            vis_util.visualize_boxes_and_labels_on_image_array(
                image,
                np.squeeze(boxes),
                np.squeeze(classes).astype(np.int32),
                np.squeeze(scores),
                category_index,
                use_normalized_coordinates=True,
                line_thickness=8,
                min_score_thresh=0.9)
            objects=[]                                       # Create objects list
            for index, value in enumerate(classes[0]):
              if scores[0, index] > 0.9:
                objects.append(scores[0, index])
                if objects[0]<0.9:
```

```
        import sys
        sys.exit("Boardingpass have not been found in post")
                                                    # Check for min 90% accuracy
    print("Accuracy Rate " + str(objects[0]))                    # Print the accuracy


    cv2.imwrite("bp_detected.jpg", image )                    # Saving detected image

    # BELOW CODE FOR SAVING DETECTED BOARDING PASS AS NEW IMAGE

    beyaz_jpg = 255*np.ones(image.shape, np.uint8)
    vis_util.draw_bounding_boxes_on_image_array(
      beyaz_jpg ,
      np.squeeze(boxes),
      color='red',
      thickness=4)
    cv2.imwrite("cevrekutu.jpg", beyaz_jpg )
    box = np.squeeze(boxes)
    min_score_thresh=0.90
    imge = Image.open("bp_detected.jpg")
    width, height = imge.size
    true_boxes = boxes[0][scores[0] > min_score_thresh]
    for i in range(true_boxes.shape[0]):
      ymin = int(true_boxes[i,0]*height)
      xmin = int(true_boxes[i,1]*width)
      ymax = int(true_boxes[i,2]*height)
      xmax = int(true_boxes[i,3]*width)

      roi = image[ymin:ymax,xmin:xmax].copy()
      cv2.imwrite("kutu0.jpg", roi)
```

```
        # BELOW CODE FOR BINARIZATION
        img = cv2.imread('kutu0.jpg',0)
        pytesseract.pytesseract.tesseract_cmd = r'C:\\Program Files\\Tesseract-
OCR\\tesseract'


        ret1,th1 = cv2.threshold(img,127,255,cv2.THRESH_BINARY)
                                                        # Global thresholding
        ret2,th2                                                        =
cv2.threshold(img,0,255,cv2.THRESH_BINARY+cv2.THRESH_OTSU)
                                                        # Otsu's thresholding
        blur = cv2.GaussianBlur(img,(5,5),0)
                                        # Otsu's thresholding with gaussian filter
        ret3,th3                                                        =
cv2.threshold(blur,0,255,cv2.THRESH_BINARY+cv2.THRESH_OTSU)
        cv2.imwrite('roi.png',th2)                      # Save binarized image

        # BELOW FOR DESKEWING THE IMAGE

      neg = 255 - th3                                   # get negative image
        angle_counter = 0                               # number of angles
        angle = 0                                       # collects sum of angles


        for line in cv2.HoughLinesP(neg, 1, np.pi/180, 325):
          x1, y1, x2, y2 = line[0]
          this_angle = np.arctan2(y2 - y1, x2 - x1)
          if this_angle and abs(this_angle) <= 10:
            angle += this_angle
            angle_counter += 1

        skew = np.rad2deg(angle / angle_counter)
```

```
rows, cols = th2.shape
rot_mat = cv2.getRotationMatrix2D((cols/2, rows/2), angle, 1.0)
result = cv2.warpAffine(th2,
                        rot_mat,
                        (cols, rows),
                        flags=cv2.INTER_CUBIC,
                        borderMode=cv2.BORDER_CONSTANT,
                        borderValue=(255, 255, 255))
cv2.imwrite('roi2.png',result)

# BELOW CODE FOR TEXT RECOGNITION

text = pytesseract.image_to_string(th2,lang=None, config='--psm 3 --oem 3')
text = text.replace("\n" , " ")
trtr = [t for t in text.split() if t.startswith("235") ]
if trtr:
   print("THY ticket found, e-mail on progress")
   print(text)
else:
   print(text)

# BELOW CODE FOR SENDING E-MAIL

import smtplib
server = smtplib.SMTP('smtp.gmail.com:587')
server.ehlo()
server.starttls()
server.login("username","password")                    # username and password
msg = "\r\n".join([
  "From: develoktay@gmail.com",                        # from sender e-mail
  "To: oktay.ekici@outlook.com",                       # to receiver e-mail
  "Subject: BoardingPass",
```

```
  "",
  text
  ])
server.sendmail("develoktay@gmail.com", "oktay.ekici@outlook.com", msg)
server.quit()
os.remove(path)
```