VIZUAL ANGLASH MASALALARİDA SUN’IY NEYTRON TO’RLARI

Nigora Qayumova Shamsiddin qizi
TerDU Amaliy matematika (sohalar bo‘yicha) mutaxassisligi magistranti
Qayumovanigora1996@gmail.com

Mamlakatimizdagi abiturentlar soni yildan-yilga oshib bormoqda. Tayyorlov kurslarini soni ham proporsional ortmoqda. Muamo test varoqasini tekshirish vaqtini kamaytirish. Masalan, bitta o'qituvchini 20 tadan o'quvchisi bo'lgan 3 yoki 4 ta guruhi bo'lsa, jami 60 yoki 80 ta o'quvchisi bo'ladi. Har haftada test sinovi o'tkazsa, uning tekshirishi uchun ko'p vaqt sarflaydi. Testlarni tanib olish orqali o'qituvchi o'z smartfoni kamerasiga test varoqasini tutish kifoya qiladi. Tanib olish algoritmini ishlash vaqti, ya'ni tanib olishga ketadigan vaqt o'rtacha 1-2 soniyani tashkil qiladi. Bu hozircha erishgan natijam. Dasturni umumiy maqsadi ta'limga yordamchi qurol bo'lishidir.

Foydalanilgan texnologiyalar tahlili

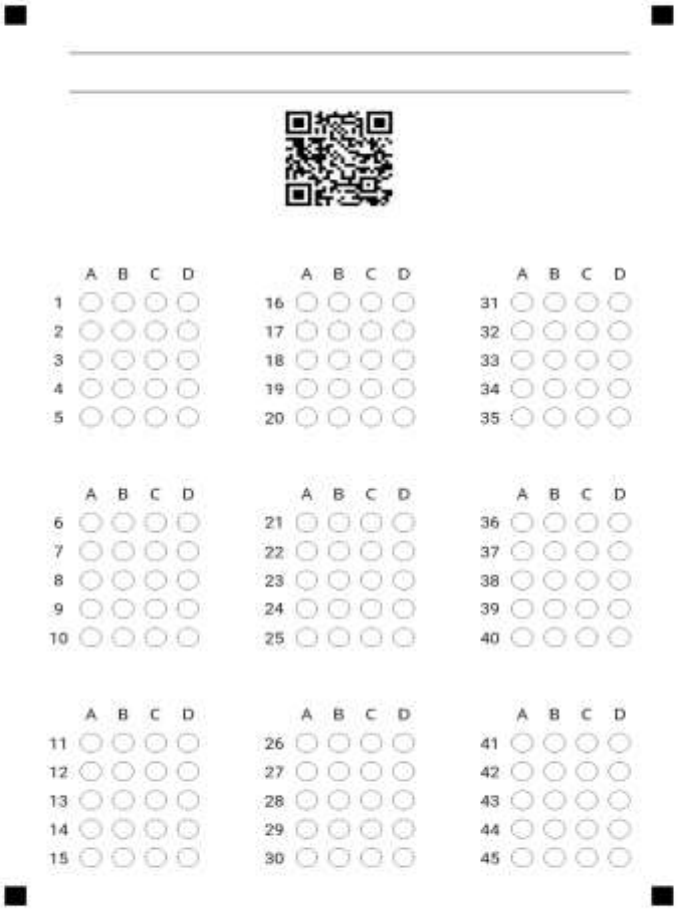
Ushbu dasturni tayyorlashda Google tomonidan ishlab chiqilgan Tensorflow platformasi, ML kit android kutubxonasi va Camera2 API texnologiyalaridan foydalanildi. Tasvirga qayta ishlov berish uchun Perspektive transform algoritmidan, bo’yalgan yuzalarni aniqlash uchun floodfill algoritmidan, tasvirni oq qora ko’rinishga o’tkazish uchun esa GrayScale algoritmidan foydalanildi.

Tensorflow – google tomonidan ishlab chiqarilgan kutubxona bo’lib u bizga sun’iy neyron tarmoqlari algoritmlaridan foydalanish imkoniyatini beradi. Men undan test blankasini tanib olish modelini tayyorlash uchun foydalandim.

ML kit- ya’ni Machine learning kit bu Mashinali o’qitishdan androidda foydalanish uchun vositalar to’plami.

Test qolipini ishlab chiqish

Tasvirlar ustida ishlash uchun avvalo uning qolipini yaratib olish muhim hisoblanadi. Ya’ni bizning algoritmimiz qanday qolipdagi tasvurlar uchun to’g’ri shuni aniqlab olishimiz kerak. Men test blankasi uchun quyidagi qolipni yaratib oldim:



Dasturni ishlash algoritmi bir necha bosqichlarni o’z ichiga oladi. Kameradan kelayotgan har bir kadr JPEG formatidagi kompressirlangan baytlar to’plamidan iborat bo’ladi. Birinchi bosqichda JPEG baytlarini bitmap matritsa ko’rinishiga keltiriladi. Tasvir bitmap ko’rinishiga o’girilganda 90 gradusga soat yo’nalishiga qarshi burilgan holatda bo’ladi. Keyin uni 0 gradusga keltiriladi. Keyingi bosqish esa qolibni tanib olish hisoblanadi. Buning uchun avval qolib tushunchasini shakllantishi kerak bo’ladi. Biz qolib sifatida 30 ta savoldan iborat test varoqasini shakllantirib oldik. Uni teng taqsimlangan 5 ta savollardan tashkil topgan 6 ta bo’lakka bo’lib oldik.

Test blankasini tanib olishda neyron tarmoqlaridan foydalanildi. Python dasturlash tilidagi Keras kutubxonasi va Tensorflow frameworkidan foydalanib o’qitib, hosil bo’lgan modelni faylga saqlab oldim.

```
import matplotlib.pyplot as plt
import matplotlib.image as img
import numpy as np
from scipy.misc import imresize
```

```
%matplotlib inline

import os
from os import listdir
from os.path import isfile, join
import shutil
import stat
import collections
from collections import defaultdict

from ipywidgets import interact, interactive, fixed
import ipywidgets as widgets

import h5py
from sklearn.model_selection import train_test_split
from keras.utils.np_utils import to_categorical
from keras.applications.inception_v3 import preprocess_input
from keras.models import load_model
```

Keyingi bosqichda esa ML kitdan foydalangan holda tayyor bo’lgan modeldan ishlatib test qolipini tanib olindi va uning tasvirdagi pozitsiyasini aniqlandi.

```
android {

    // ...

    aaptOptions {
        noCompress "tflite"  // Your model's file extension: "tflite", "lite", etc.
    }
}
```

```
val remoteModel = FirebaseCustomRemoteModel.Builder("your_model").build()
```

```
val conditions = FirebaseModelDownloadConditions.Builder()
    .requireWifi()
    .build()
FirebaseModelManager.getInstance().download(remoteModel, conditions)
    .addOnCompleteListener {
        // Success.
    }
```

```
root_dir = 'food-101/images/'
rows = 17
```

```
cols = 6
fig, ax = plt.subplots(rows, cols, frameon=False, figsize=(15, 25))
fig.suptitle('Random Image from Each Food Class', fontsize=20)
sorted_food_dirs = sorted(os.listdir(root_dir))
for i in range(rows):
    for j in range(cols):
        try:
            food_dir = sorted_food_dirs[i*cols + j]
        except:
            break
        all_files = os.listdir(os.path.join(root_dir, food_dir))
        rand_img = np.random.choice(all_files)
        img = plt.imread(os.path.join(root_dir, food_dir, rand_img))
        ax[i][j].imshow(img)
        ec = (0, .6, .1)
        fc = (0, .7, .2)
        ax[i][j].text(0, -20, food_dir, size=10, rotation=0,
                ha="left", va="top",
                bbox=dict(boxstyle="round", ec=ec, fc=fc))
plt.setp(ax, xticks=[], yticks=[])
plt.tight_layout(rect=[0, 0.03, 1, 0.95])
```

Test javoblarini tekshirish

Test javoblarini tekshirish uchun avvalo RGB ranglar to'plamidan GRGBga o'girib olindi. Tezlikni yanada oshirish uchun bu bosqichga ozroq o'zgartirish kiritildi. Ya'ni butun tasvirni o'girmasdan kerakli nuqtanigina o'girish yetarli bo'ladi. GRGB bu GrayScale hisoblanadi. RGB tasvir quyidagi formula orqali GRGB ga o'giriladi:

Grayscale = 0.299 * R + 0.587 * G + 0.114 * B

Foydalanilgan adabiyotlar

1. A.Maraximov, T.Akramov Sun'iy o'rganish asoslari(Machine learning) T-2020
2. *Pratap Dangeti, Allen Yu, Claire Chung, Aldrin Yim, Theodore Petrou* Numerical Computing with Python *Birmingem-2018*
3. https://insights.stackoverflow.com/survey/2021#technology