

Az issue alapján a következő módosításokat hajtottam végre a kódon belül:

- #1. létrehoztam egy config fájlt, amelyben az egyes kategóriák vannak megadva, ezekkel dolgozik majd a program.
 - #2. a keresés ciklusban történik egy, a felhasználó által megadott mappa elemeit vizsgálva.
-

#1.

Az eddigi beolvas() függvényt módosítottam, az új állapotában a config fájlnevet fogadja (config.txt) az eddigiektől eltérő konkrét kategóriák megadása helyett. A config fájl ilyen formában áll rendelkezésre:

```
Hamburger hamburger/  
Stop stop/  
Mobil mobil/
```

A kategória nevét és a hozzá tartozó mappát tabulátor karakter választja el egymástól. A beolvasás során a fájl tartalma ellenőrzésen esik át, ami azt jelenti, hogy csak a meglévő kategóriáknak megfelelő soronkénti első tag elfogadott. Az összetartozó értékpárokat egy map konténerben tároltam el, ahol a kulcs maga a kategória (string), az érték pedig a képeket tartalmazó mappa (string).

A fájl nyitása az ifstream osztály segítségével történik meg. Ha a fájl nyitása sikertelen, akkor a beolvas függvény false értékkel tér vissza, ennek következménye, hogy a main-ben a további futás (keresés például) nem fog majd lefutni, mert nem áll rendelkezésre az összehasonlítás alapja. Ha a nyitás sikeres volt, megkezdődhet az egyes sorok vizsgálata is. Minden beolvasott sor a tabulátor karakternél két részre lesz bontva, ezek a tagok kerülnek majd a map-be. A daraboláshoz a string substr metódusát használtam fel. Az egyes kategóriák betöltése után ennek megfelelő üzenet jelenik meg a konzolon.

```
Hamburger betoltese sikeres volt!  
Stop betoltese sikeres volt!  
Mobil betoltese sikeres volt!
```

Olyan esetben, amikor nem megfelelő a kategória vagy a sor formátuma nem az elvárt, azonnal hibaüzenet jelenik meg. Például az Auto auto/ sorra a program hibaüzenetet fog adni.

```
Ez a sor nem megfelelo: Auto    auto/
```

Az egyes kulcs-érték párok ekkor már rendelkezésre állnak, de még maguk a képek nem lettek beolvasva. A map-et végig járva az egyes kategóriáknak megfelelően először a pozitív, majd a negatív minták kerülnek vectorokba. Az értékeket a map second adattagjával tudjuk lekérni, valamint ehhez hozzá fűzve a fájlok neveit, valamint a kiterjesztést, rendelkezésre állnak a hivatkozások az egyes fotókra, ezáltal a tárolókba kerülhetnek.

A negatív minták feldolgozása hasonló módon történik meg. Annyi változtatást eszközöltem, hogy nem használok külön negatív képeket, mint eddig, hanem az aktuálisan pozitív kategóriához keresek egy másik objektum kategóriát, pl. ha a

hamburgerek kerülnek beolvasásra pozitívként, akkor ehhez a mobil kategória megfelelők negatív mintaként. Az eddigiekben minden egyes kategóriára ugyan azok a negatív képek lettek felhasználva. Azt, hogy melyik legyen a negatív kategória egy random szám generálás dönti el. Természetesen az nem fordulhat elő, hogy a hamburgerekhez a pozitív és a negatív minták is egységesen hamburgerek. A véletlen számot addig kell generálni, amíg előfordul olyan eset, amikor eltérés tapasztalható, ekkor megfelelően előáll az egyes kategóriákra a tároló:

0. elem: burger/burger00.jpg	Pozitív? igen
1. elem: burger/burger01.jpg	Pozitív? igen
2. elem: burger/burger02.jpg	Pozitív? igen
3. elem: burger/burger03.jpg	Pozitív? igen
4. elem: burger/burger04.jpg	Pozitív? igen
5. elem: burger/burger05.jpg	Pozitív? igen
6. elem: burger/burger06.jpg	Pozitív? igen
7. elem: burger/burger07.jpg	Pozitív? igen
8. elem: stop/stop00.jpg	Pozitív? nem
9. elem: stop/stop01.jpg	Pozitív? nem
10. elem: stop/stop02.jpg	Pozitív? nem
11. elem: stop/stop03.jpg	Pozitív? nem
12. elem: stop/stop04.jpg	Pozitív? nem
13. elem: stop/stop05.jpg	Pozitív? nem
14. elem: stop/stop06.jpg	Pozitív? nem
15. elem: stop/stop07.jpg	Pozitív? nem

#2.

A keresés az adathalmazban egy teszt nevű mappában rendelkezésre álló képeken történő végig járást jelenti. A teszt mappa elérhetősége nem a config fájlban kapott helyet. Az előző verziókban a program elején a felhasználó 1 képet adott meg, amely össze lett hasonlítva a címkézett mintákkal. Ezt a bekérést mozgattam át a program végére kis módosításokkal. Nem megfelelő útvonal esetén hibaüzenet jelenik meg.

```
Kerem adjon meg egy letezo utvonalat a kereseshez
-->> teszt2
Nem letezo utvonal!
```

Ha létező útvonalat adtak meg, akkor a mappában található képek elemzése egyenként megtörténik, majd az eredmény visszaadásra kerül a megfelelő kiíratással együtt. A teszt/ mappában ezek az elemek találhatóak meg: teszt00.jpg (hamburger), teszt01.jpg (stoptábla), teszt02.jpg (mobil), teszt03.jpg (hamburger).



teszt00.jpg



teszt01.jpg



teszt02.jpg



teszt03.jpg

```
Kerem adjon meg egy letezo utvonalat a kereseshez
-->> teszt/

Az objektum (teszt/teszt00.jpg) egy hamburger!

Az objektum (teszt/teszt01.jpg) egy stoptabla!

Az objektum (teszt/teszt02.jpg) egy mobiltelefon!

Az objektum (teszt/teszt03.jpg) egy hamburger!
```

Előfordulhat olyan eset, amikor nincs találat, ekkor „Az objektum nem meghatározható” felirat fog megjelenni. Ez megeshet akkor, ha nem esik egyik kategóriába sem a vizsgált kép, valamint akkor is, ha bele esik. Tekintettel a kis számú tanító adatra, csak a pozitív mintákhoz képesti nagy egyezőséget mutató fotókon kapunk várt eredményt.