

**Elektrotehnički fakultet - Univerzitet u Beogradu**  
**Katedra za signale i sisteme**

**Neuralne mreže (13S053NM)**  
**Prvi projektni zadatak**



**Mihajlo Dražić 2020/0656**

**Bora Miletić 2020/0319**

## 1. Postavka projekta

Cilj projekta je pravljenje konvolucione neuralne mreže, sa ciljem da se vrši klasifikacija zadatog dataset-a u predodređene klase. Naš dataset je podeljen u 6 klasa i to su buildings(zgrade), forest(šuma), glacier (glečer), sea(more) i street(ulica).

Odabrani skup podataka sadrži oko 14 hiljada slika dimenzija 150x150. Za početak skaliramo sve slike na dimenzije 75x75. Potom ih delimo na dva skupa, Xtrain(trening skup) i Xval(validacioni skup) uz pomoć funkcije **image\_dataset\_from\_directory** iz keras biblioteke.

Nakon podele, prikazujemo random 10 slika da bi imali blagi uvid u to kakve su slike.



## 2. Data augmentation(veštačko povećavanje dataset-a)

Data augmentation je veštačko dodavanje podataka upotrebom neki filtera iz biblioteke keras . Mi smo upotrebili random flip(horizontalno okretanje slika), random zoom(zumiranje slika) i random rotate (rotacija slika). Ovim smo višestruko povećali dataset, pa je zato obučavanje znatno duže, ali je bolje zato će se mreža bolje obučiti ako ima više podataka.

```
from keras import layers
from keras import Sequential
data_augmentation = Sequential(
    [
        layers.RandomFlip("horizontal", input_shape=(img_size[0], img_size[1], 3)),
        layers.RandomRotation(0.25),
        layers.RandomZoom(0.1),
    ]
)
```

Ovde možemo videti kako izgleda jedna slika planine nad kojom je primenjen data augmentation.



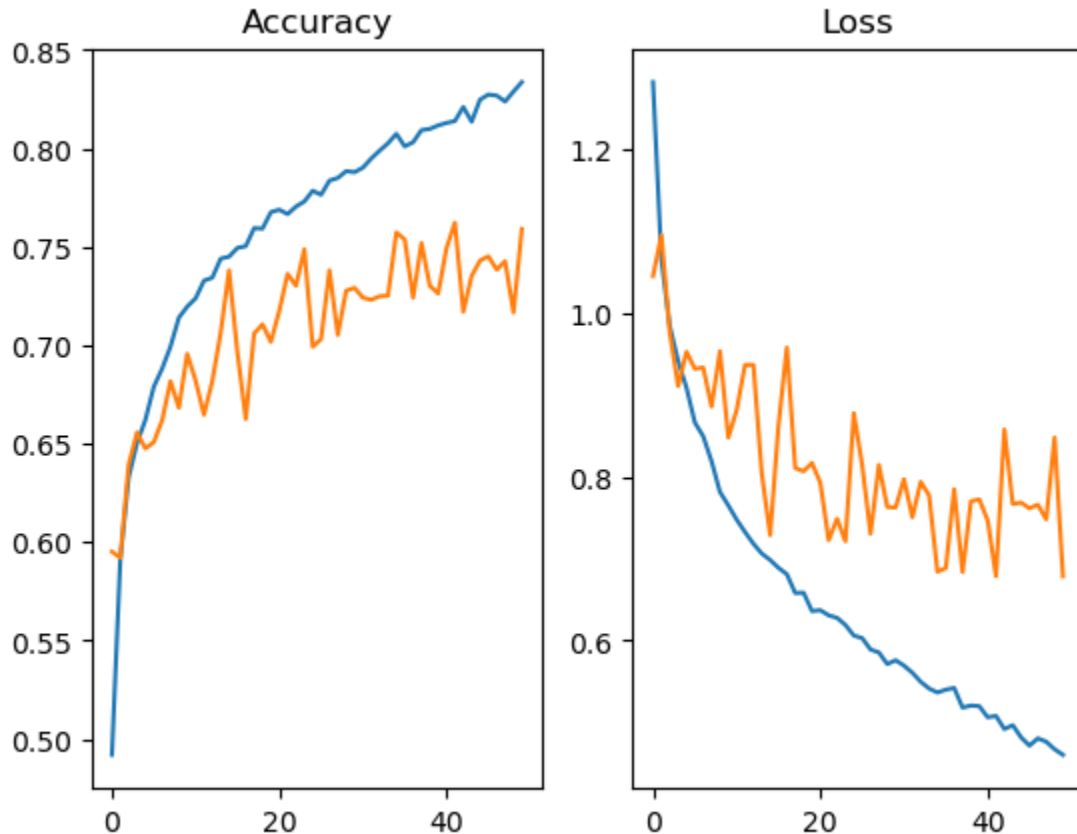
### 3. Pravljenje neuralne mreže

Kada smo veštački dodali slike u dataset, možemo preći na najvažniji deo projekta, a to je kreiranje samo neuralne mreže. Dodajemo konvolucione slojeve i nakon njih maxpooling slojeve 3 puta. Zatim dodajemo dropout sloj koji služi kao zaštita od preobučavanja(kasnije dodajemo i early stopping). Nakon toga dodajemo dva dense("gusta") sloja sa aktivacionim funkcijama relu i softmax. Ta dva sloja služe da prikupe sve podatke iz predhodnih slojeva i dođu do zaključka o kojoj klasi se radi.

```
model = Sequential([
    data_augmentation,
    layers.Rescaling(1./255, input_shape=(75, 75, 3)),
    layers.Conv2D(16, 3, padding='same', activation='relu'),
    layers.MaxPooling2D(),
    layers.Conv2D(32, 3, padding='same', activation='relu'),
    layers.MaxPooling2D(),
    layers.Conv2D(64, 3, padding='same', activation='relu'),
    layers.MaxPooling2D(),
    layers.Dropout(0.2),
    layers.Flatten(),
    layers.Dense(128, activation='relu'),
    layers.Dense(num_classes, activation='softmax')
])
```

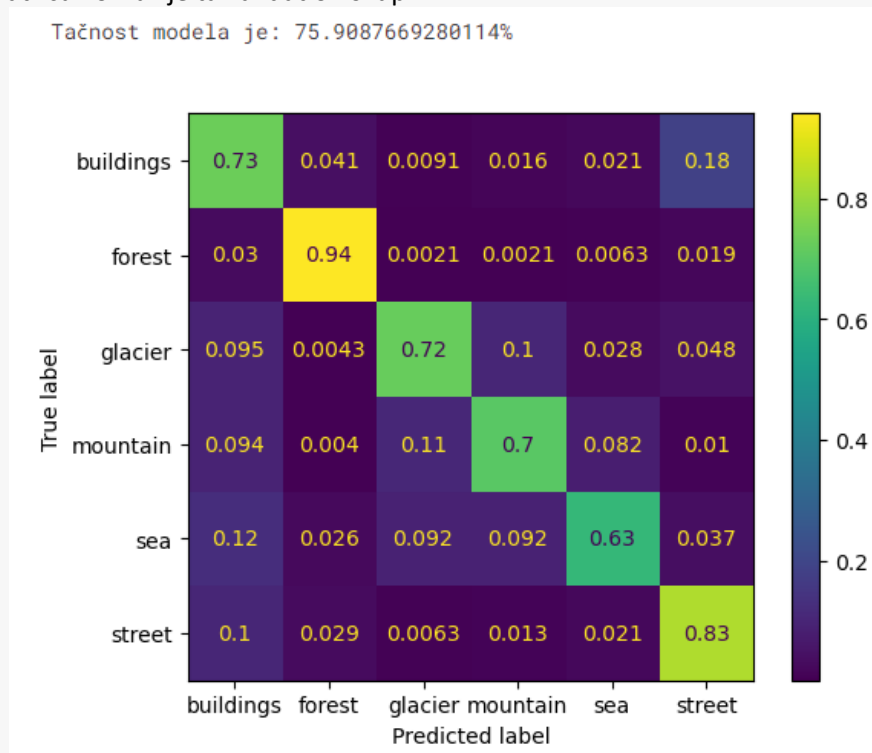
#### 4. Praćenje parametar na izlazu

Da bi pratili parametre na izlazu, koristimo funkcije accuracy i loss funkciju i to stavljamo na grafike. Vidimo da je mreža relativno prihvatljive tačnosti. Problem je u tome što bi bilo korisno koristiti celokupne slike, jer se na nekim slikama nalaze delovi drugih klasa, npr. na ulici se nalaze zgrade, na planinama može biti snega pa da liči na glečcer itd. Nažalost, zbog ograničenih hardverskih resursa, postavljene dimenzije su maksimalne koje računar na kome je rađen projekat može da podrži za toliku količinu podataka.

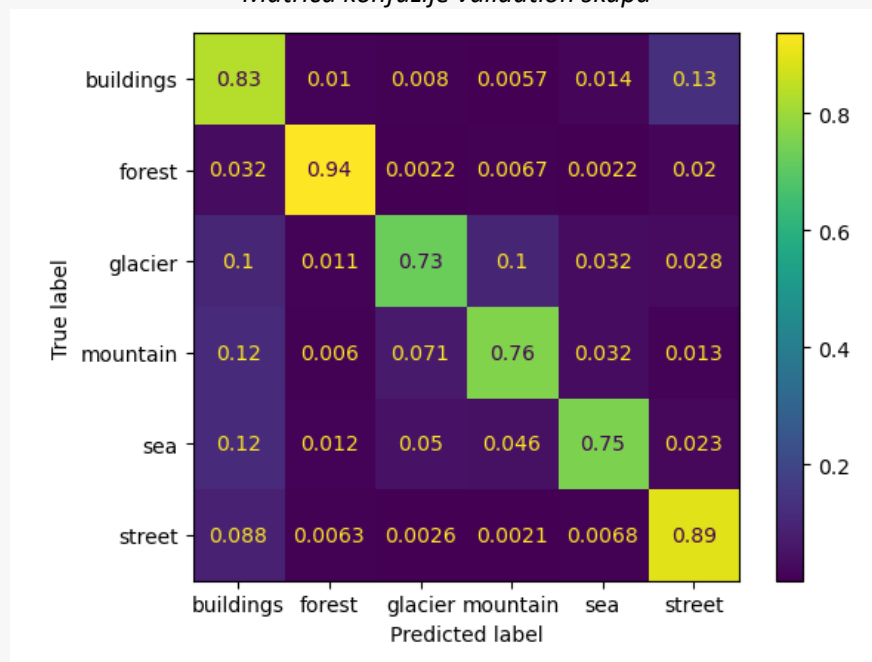


## 5. Konačna analiza rezultata

Najbolji uvid u rezultate možemo videti u matricama konfuzije. Takođe, ukupnu tačnost možemo videti u z pomoć funkcije `accuracy_score` i vidimo da je tačnost 75,9% što je vrlo zadovoljavajuće, s obzirom na to da bi i čoveku ovakva klasifikacija mogla da zadaje probleme za određene slike. Ispod vidimo matricu konfuzije za validation skup.



Matrica konfuzije validation skupa



Matrica konfuzije train skupa

Primećujemo da je najveća greška razlikovanje zgrada(buildings) i ulice(street) zbog već pomenutog problema da se na ulici mogu naći zgrade, o čemu bi mogla da se vodi polemika šta se smatra samo ulicom. Takođe sličnu gresku vidimo i planinu(mountain) i glečer(glacier), gde je problem to što se na planini može naći sneg i led koji liče na glečer, a i glečer može imati oblik planine.

U nastavku su slike ulica, koju su pogrešno klasifikovane kao zgrade, ali to nije toliko problem neuralne mreže, jer bi i čovek mogao dvosmisleno da tumači te slike.

