Error Lego

A képen játék látható

Automatikusan generált leírásA képen ülő, asztal, szoba, nagyon nagy látható

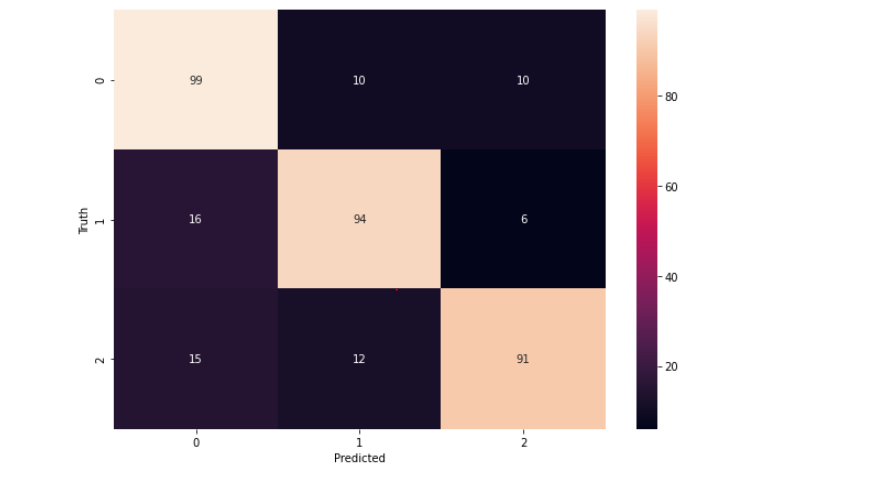
Automatikusan generált leírásA képen beltéri, ülő, kicsi, asztal látható

Automatikusan generált leírás

Feladat ennek a 3 fázisnak a megkülönböztetése/felismerése.

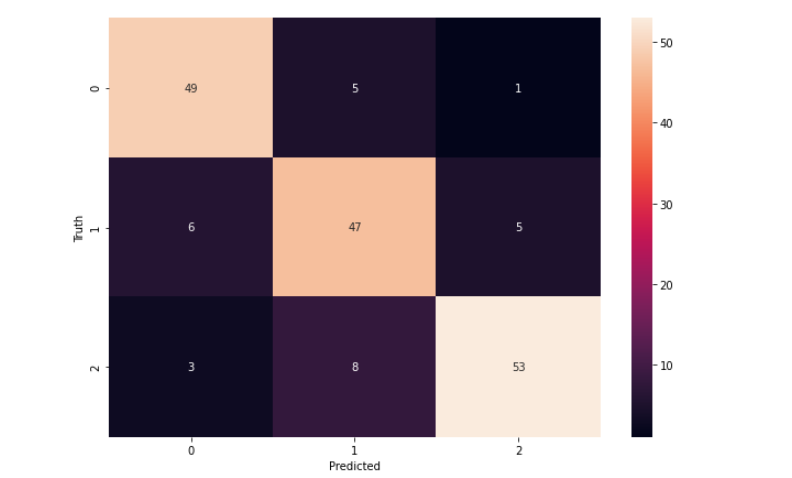
***RandomForestClassifier***: Először beolvassuk a képeket még nem színesen és megnézzük, hogy teljesít a model.

**Először a test\_size = 0.2**: 0.8-as pontosságot ért el a model.

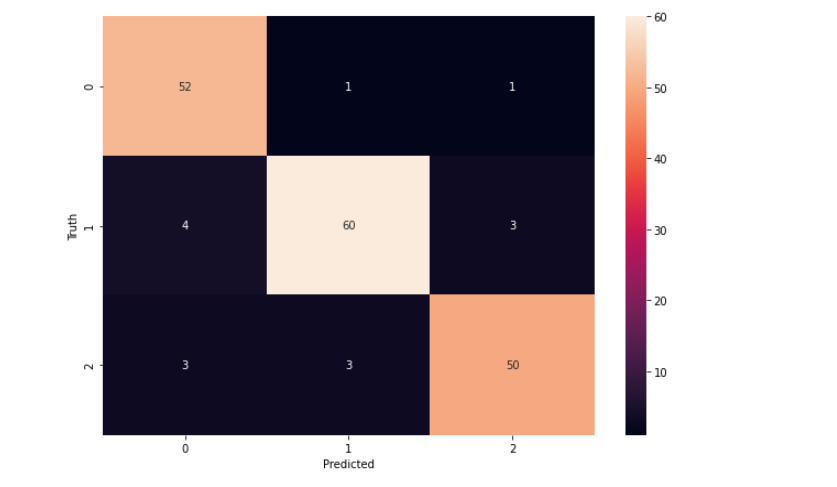


Nem olyan rossz eredmény, de nagyobb pontosság szükséges.

**test\_size = 01**: 0.84 így jobban teljesített a model:



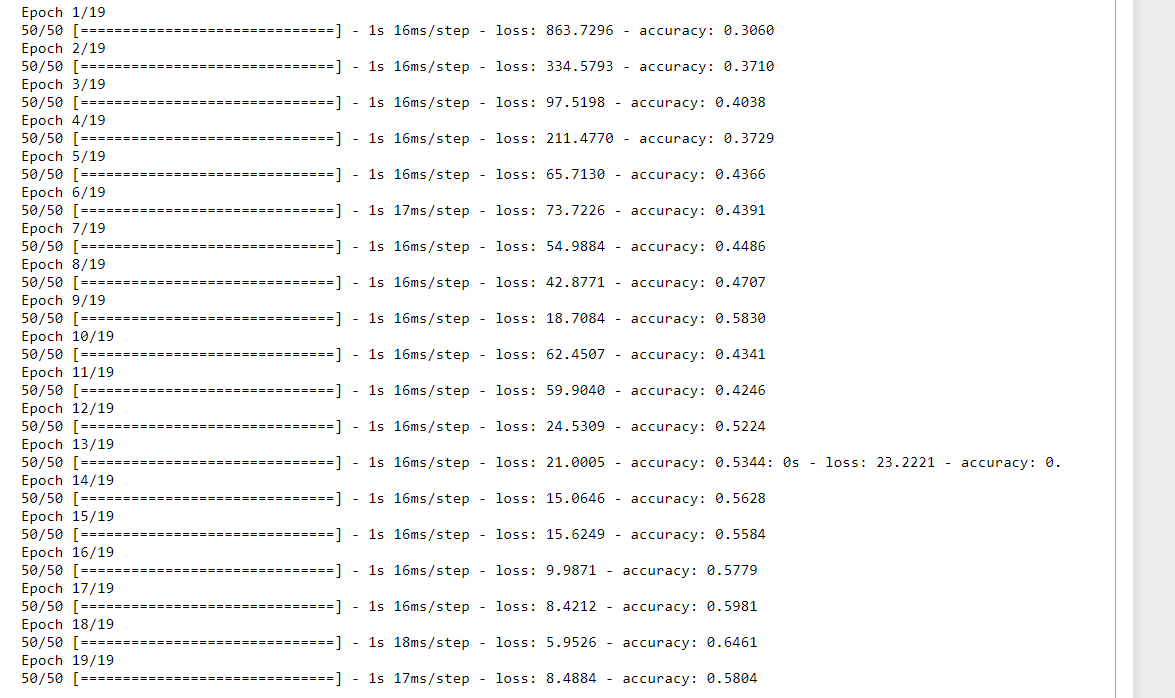
**Most színesben olvassuk be a képeket (**így más jóval hosszabb a beolvasás**):** úgy gondolom, hogy így pontosabb kellene, hogy legyen a model mivel így talán élesebbek a különbségek.  
**Hála Istennek** igaz volt a sejtés így a model már 0.915 pontosságot ért el.



**TensorFlow\_Kreas:**

**Sequential model:**

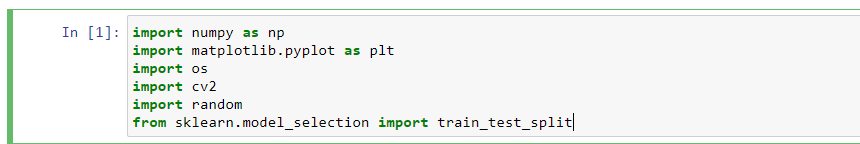
Először egy egyszerű neurális hálót próbáltam ki csak Dense Layer-ek el de ezek nem hoztak jó megoldást ezért további Layer-ekre van szükség a pontosság növeléséhez. Most a pontosság csak 0.64 maximum.



Convolutional neural network tensorflow:

Itt kétdimenziós konvolúciós neurális hálót használtam. Meg is látszik az eredményen mivel a végére a model elérte a 1-es pontosságot.

Magyarázat: Először is a szükséges importok a programhoz:

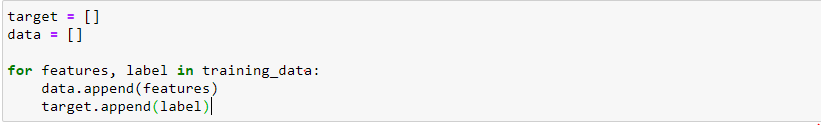


Ezekre szükség van a képek beolvasásához, összekeveréséhez és train és test adatok sztválasztásához.

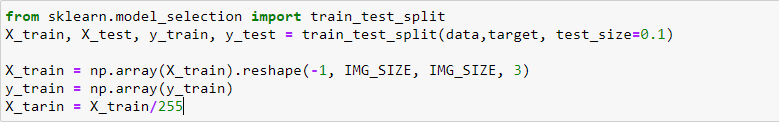
Következő lépésben beolvassuk a képeket színesben és közben átméretezzük 100X100-as méretűre mert ez is elég jó minőségű ahhoz hogy a model meg tudja különböztetni.



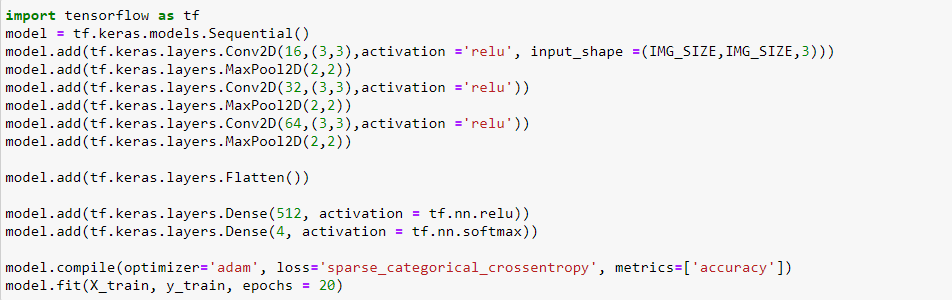
Láthatjuk kiírva, hogy az egyes kategóriákról hány kép van. Ezek után összekevertem az adatokat, hogy ne sorban legyen mert akkor a model azt tanulja meg hogy például mindig csak 1-es kategória utána meg azt hogy mindig csak 2-es stb.



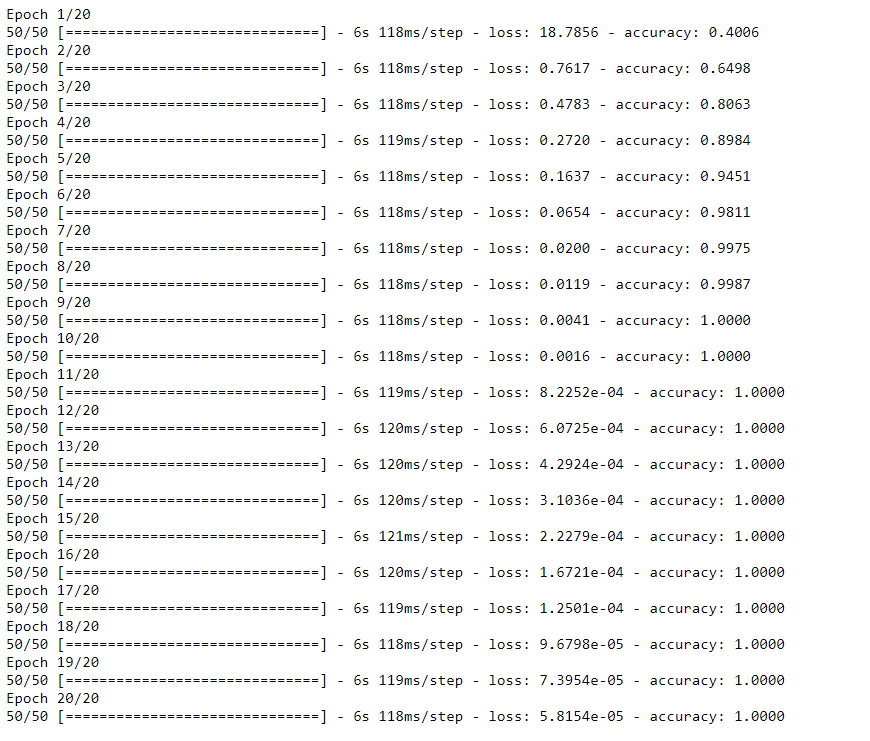
Itt valósult meg az adataink szétválasztása külön data és target-re. Ezeket szétválasztottam külön teszt és train adatokra.



Utána a két train adatot reshape-eltem mert a modelünknek megfelelő alakú numpy tömbre van szüksége. Utána az X\_train tömböt osztottam 255-el, hogy benne csak 0->1-ig legyenek értékek mert így a model könnyebben és jobban tanul belőle. És ezután következett a neurális háló megalkotása.



Az első Layer-nél meg kell adni az adat formáját vagyis az imput\_shape-et ami itt ugye a kép méret szor a kép méret és utána van egy 3 azért mert színes képekről van szó és így egy pixelnek 3 értéke van: red green blue. Itt ’adam’ optimizert használtam és a loss ’categorical’ mivel három fázisunk van.



**Hála Istennek** elég szép eredményeket produkált a model már a kilencedik epoch-nál láthattuk, hogy eléri az 1-es pontosságot, ami a 100%. És utána sem romlott el.

RMSprop optimizer:

Most kipróbáltam egy másik optimizert ami széntén jó eredményeket hozott de 20 epoch alatt maximum 0.98 volt a pontossága szóval kicsit rosszabbúl teljesített mint az Adam. 