

Documentarea proiectului

SVM:

SVM (Support Vector Machines) este un model de machine learning supervizat din librăria sklearn. Acesta are ca principale avantaje versatilitatea (diverse funcții kernel pot fi folosite ca și funcții de decizie, inclusiv funcții custom) și eficiența pe spații cu multe dimensiuni (în cazul nostru pixelii în format 2D din poză, puși ca și triplete RGB). Pe de altă parte, are ca principal punct slab problemele în care numărul de features este mult mai mare decât cel de samples. Pentru tabelul și matricea de confuzie de mai jos, am folosit clasa de clasificare SVC (Support Vector Clustering).

Variabila C atribuită modelului	Acuratețea pe validare obținută
C = 0.1	46%
C = 1	54%
C = 5	56%
C = 10	55%

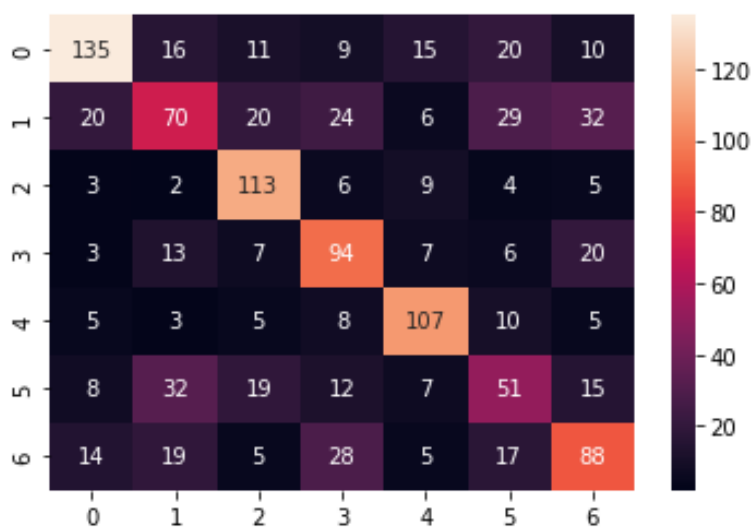


Fig.1 Matricea de confuzie pentru modelul SVC cu acuratețea maximă (C = 5)

SVC-ul cu $C = 5$ a obținut pe Kaggle 0.56534 pe testele publice.

CNN:

CNN (Convolutional Neural Network) este o clasă de rețea neurală, specializată mai repede pe învățarea directă pe date ce pot fi expuse sub formă de tabele sau matrice, principalul tip de date care poate fi expus astfel fiind o imagine.

Principalul punct forte al rețelelor neurale convoluționale este recunoașterea de locuri, obiecte, fețe, persoane din imagini. Este deasemenea eficientă și la prelucrarea altor tipuri de date decât poze, cum ar fi fișierele audio. Necesitatea unor tipuri de date subformă de tabele minim 2 dimensionale poate fi considerat principalul dezavantaj al rețelelor convoluționale, neputând fi aplicabil pe majoritatea problemelor de procesare de limbaj natural.

Layer (type)	Output Shape	Param #			
			dropout_32 (Dropout)	(None, 1, 1, 128)	0
conv2d_15 (Conv2D)	(None, 15, 15, 32)	416	flatten_5 (Flatten)	(None, 128)	0
max_pooling2d_15 (MaxPoolin g2D)	(None, 7, 7, 32)	0	dense_20 (Dense)	(None, 256)	33024
batch_normalization_30 (Bat chNormalization)	(None, 7, 7, 32)	128	batch_normalization_33 (Bat chNormalization)	(None, 256)	1024
dropout_30 (Dropout)	(None, 7, 7, 32)	0	dropout_33 (Dropout)	(None, 256)	0
conv2d_16 (Conv2D)	(None, 6, 6, 64)	8256	dense_21 (Dense)	(None, 128)	32896
max_pooling2d_16 (MaxPoolin g2D)	(None, 3, 3, 64)	0	batch_normalization_34 (Bat chNormalization)	(None, 128)	512
batch_normalization_31 (Bat chNormalization)	(None, 3, 3, 64)	256	dropout_34 (Dropout)	(None, 128)	0
dropout_31 (Dropout)	(None, 3, 3, 64)	0	dense_22 (Dense)	(None, 64)	8256
conv2d_17 (Conv2D)	(None, 2, 2, 128)	32896	batch_normalization_35 (Bat chNormalization)	(None, 64)	256
max_pooling2d_17 (MaxPoolin g2D)	(None, 1, 1, 128)	0	dropout_35 (Dropout)	(None, 64)	0
batch_normalization_32 (Bat chNormalization)	(None, 1, 1, 128)	512	dense_23 (Dense)	(None, 7)	455
			=====		
			Total params: 118,887		
			Trainable params: 117,543		
			Non-trainable params: 1,344		

Fig.2 Arhitectura rețelei mele convoluționale

Learning rate: 1e-3

Optimizator: SGD cu momentum = 0.9

Acuratete pe validare: 53%

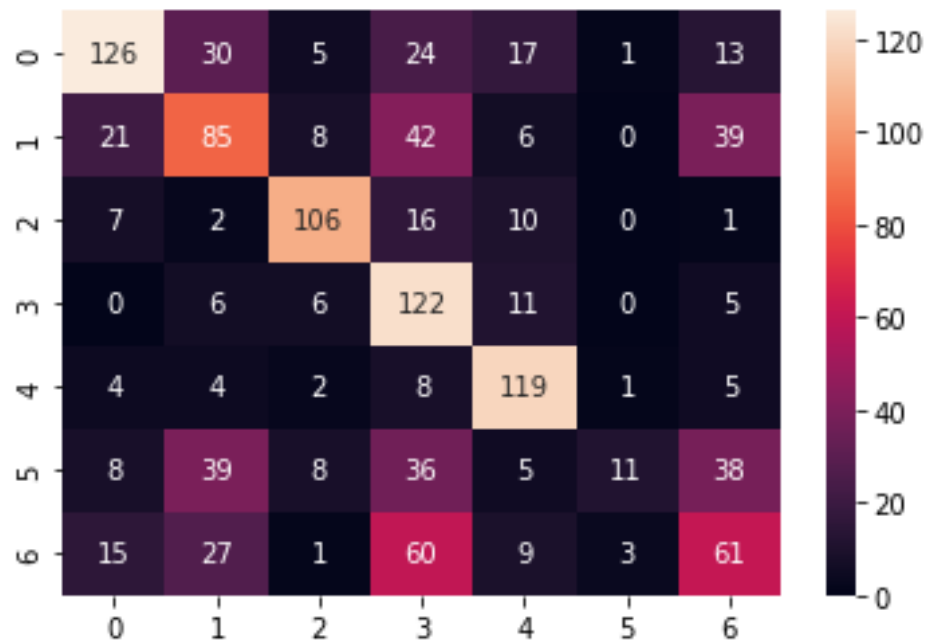


Fig.3 Matricea de confuzie pentru modelul CNN cu optimizatorul SGD

Learning rate: 1e-3

Optimizer: Adam

Acuratete pe validare: 63%

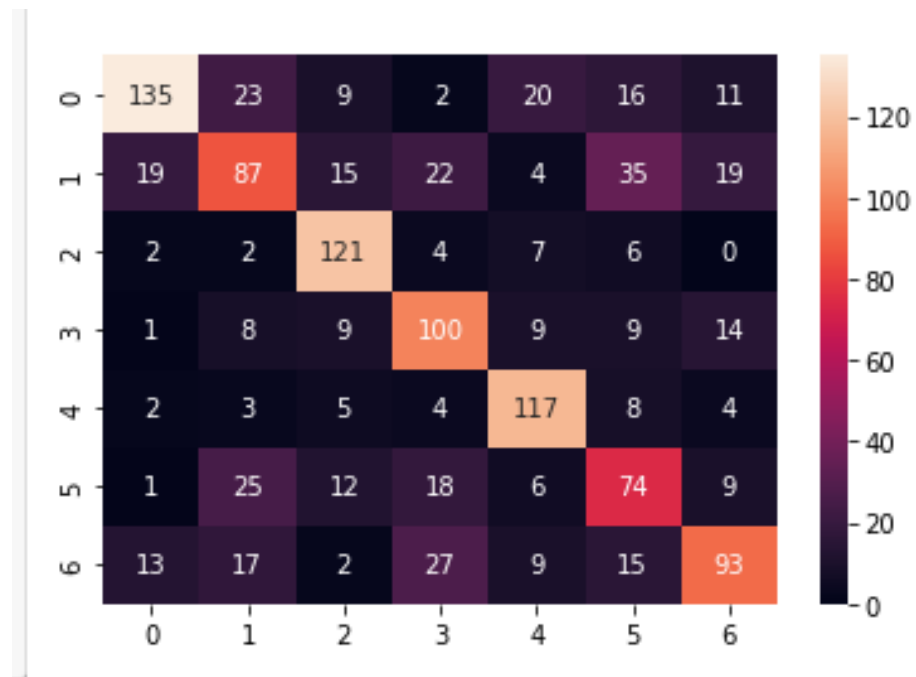


Fig.4 Matricea de confuzie pentru modelul CNN cu optimizatorul Adam

CNN definită în figura 4 (și cu arhitectura din figura 2) a reușit să obțină 0.65340 pe Kaggle pe testele publice.

EfficientNet:

EfficientNet este un model bazat pe un o rețea neurală convoluțională, pre-antrenată pe o bază vastă de imagini (ImageNet). EfficientNet obțin de regulă eficiență și acuratețe semnificativ mai mare decât rețelele convoluționale clasice.

EfficientNet is a convolutional neural network architecture and scaling method that uniformly scales all dimensions of depth/width/resolution using a compound coefficient. Singurul dezavantaj notabil al acestei rețele, comparativ cu un CNN, este că dacă train-set ul nostru este **foarte mare**, la un anumit punct preantrenarea poate să modifice în rău prezicerile corecte pe care le-ar fi avut modelul doar cu antrenarea pe train-set, însă vorbim de un train-set așa de variat și de mare încât, singur, să obțină o acuratețe de peste 90-95%, deci și acolo, diferențele în favoarea CNN ului ar fi neglijabile. Am folosit în submisiile mele versiunile EfficientNet-ului de la B0 la B4.

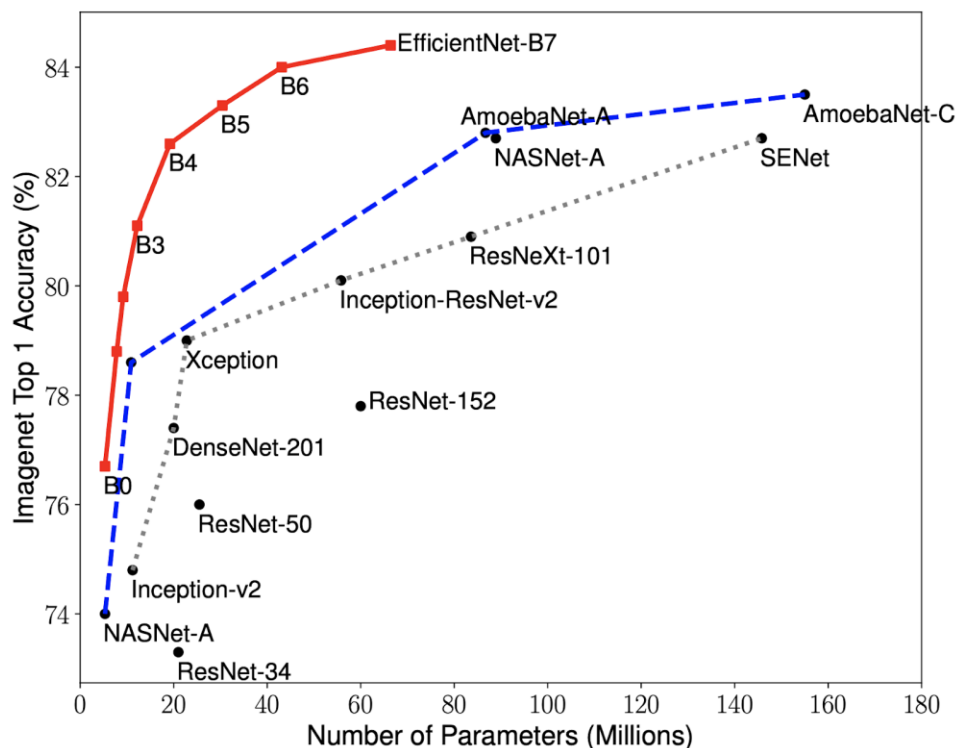


Fig.5 Acuratețea modelelor EfficientNetB0-B7 comparativ la alte modele pre-antrenate existente

Learning rate: 1e-4

Optimizator: Adam

Acuratete pe validare: 69%

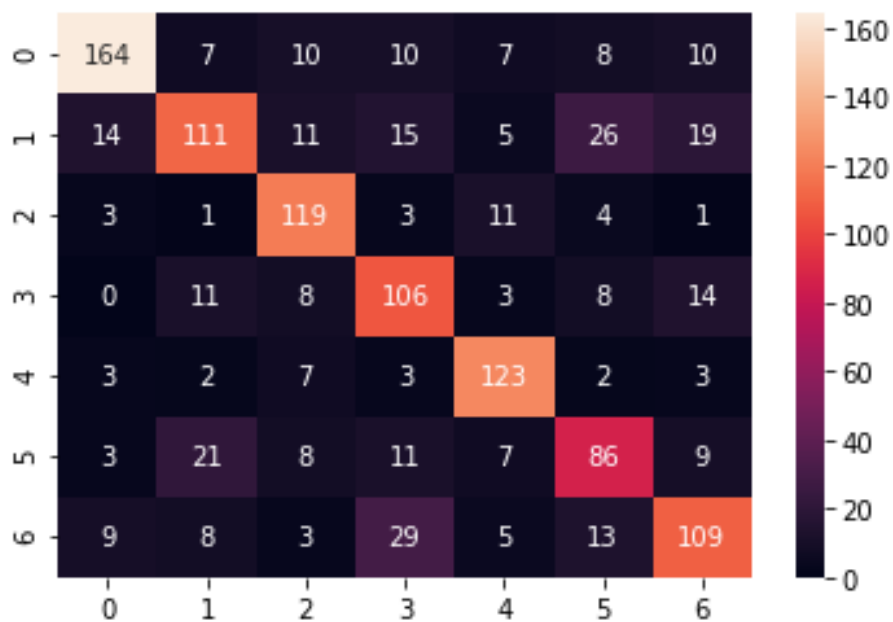


Fig.6 Matricea de confuzie pentru modelul EfficientNet (optimizator Adam)

Algoritmul cu EfficientNet definit anterior, cu matricea de confuzie de la figura 6 a obținut pe Kaggle scorul 0.69744.

TransferLearning

Conceptul de transferlearning reprezintă folosirea unei rețele pre-antrenate pe care **o antrenăm la rândul ei** și cu train-set ul nostru. Rețeaua EfficientNet este preantrenată de pe ImageNet, o bază de imagini care stă la baza modelului

Vot majoritar

Definim votul majoritar dintre mai multe modele, generarea unei predicții noi ca fiind **cea mai precisă clasă**, de către modele alese, pentru fiecare imagine din fișierul de train. În cazul în care modele au predicții distincte 2 câte 2 pentru o imagine X, o să alegem predicția modelului care a obținut rezultatul cel mai bun pe datele de validare ca răspuns.

Pentru scorul maxim obținut pe Kaggle (0.71022) am efectuat un **vot majoritar** asupra mai multor submisii cu acuratețea între 68% și 69% (generate folosind tot tehnica de transfer-learning utilizând rețele preantrenate cum ar fi EfficientNetb0, EfficientNetb1, EfficientNetb2, EfficientNetb3, EfficientNetb4, ResNet34).