

**Colegiul National "Liviu Rebreanu" Bistrita**

**Lucrare de atestat 2020**

**Clasificarea imaginilor folosind Retelele  
Neurale Convolutionale implementate cu  
Keras si Python**

Profesor coordonator: Galatan Constantin

Autor: Buzan Dan-Alexandru Cls. a XII – a A

# Cuprins

1. Prezentarea limbajului Python si a bibliotecii TKinter	3
2. Ce sunt retelele neurale convolutionale ?	3
3. Proiectarea arhitecturii unei retele neurale. Componente.	5
4. Antrenarea, validarea, si testarea retelei.	8
5. Probleme intampinate. Imbunatatiri viitoare	10
6. Bibliografie	10

## **1.Prezentarea limbajului Python si a bibliotecii TKinter**

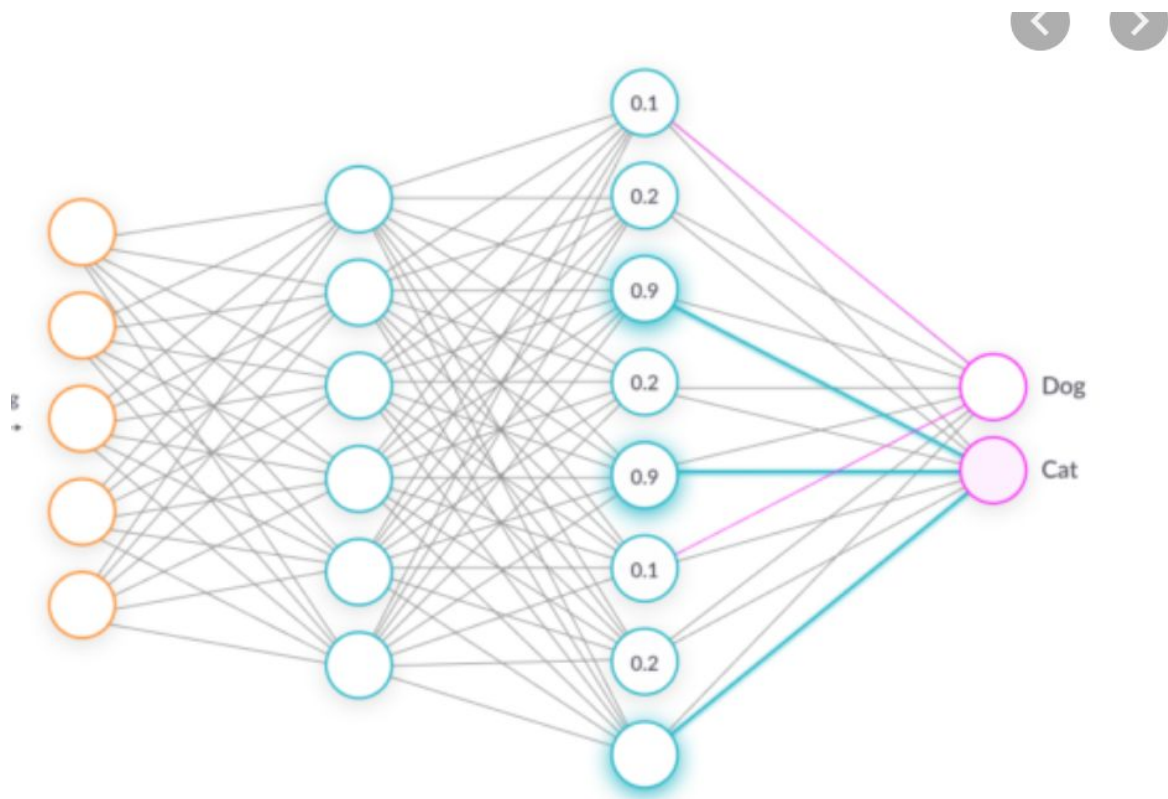
Pentru acest proiect am ales limbajul Python si mediul integrat PyCharm. Python este un limbaj high-level interpretat, cunoscut pentru usurinta si rapiditatea cu care se pot implementa solutii software, dar si pentru multitudinea de biblioteci de care dispune.

TKinter este biblioteca GUI standard a limbajului Python, acest lucru venind cu avantaje si dezavantaje. Unul dintre avantaje ar fi faptul ca biblioteca este inclusa in instalarea Python, nefiind nevoie de alte instalari. Pe de alta parte, biblioteca este considerata destul de primitiva, neavand prea multe facilitati moderne ca si alte biblioteci(Kiwi, PyQt etc). Eu am ales Tkinter pentru usurinta de folosire, neavand nevoie de un GUI foarte avansat, scopul proiectului fiind acumularea de cunostiinte in domeniul invatarii automate.

Implementarea si antrenarea rețelei neurale am facut-o in mediul Jupyter Notebook, fiind mai usor pentru manipularea unor baze mari de date, dar si pentru antrenarea modelelor in Cloud(Kaggle, Google Colab)

## **2.Ce sunt rețelele neurale convolutionale ?**

Retelele neurale convolutionale(CNN- Convolutional Neural Networks) sunt un anumit tip de rețele neurale folosite in invatarea profunda(deep learning). Aceste rețele pot fi vizualizate ca niste noduri a unui graf, ele primind datele de intrare(pixelii unei imagini in cazul de fata) si afisand probabilitatea ca imaginea sa apartina unei clase de obiecte(in cazul nostru, fructe si legume). Intre nodurile de intrare si cele de iesire pot exista mai multe straturi de noduri, numite noduri ascunse. Fiecare strat de noduri sunt legate de nodurile din stratul anterior si cele din stratul urmator, si pe baza anumitor functii matematice, ofera niste date nodurilor in stratul urmator in functie de datele primite de la nodurile din stratul anterior.



Acesta este un exemplu de o retea neurala care decide daca intr-o poza se afla un caine sau o pisica. In ultimele noduri se calculeaza probabilitatea ca in poza sa se afle un caine, respectiv o pisica, si decizia finala se ia in functie de valoarea mai mare.

Lucrul care individualizeaza retele neurale convolutionale fata de alte retele neurale sunt straturi speciale de matrici convolutionale, explicate mai jos.

### **3.Proiectarea arhitecturii unei retele neurale. Componente.**

Problema pe care am abordat-o este clasificarea unei imagini in peste 120 de clase care contin fructe si legume. Setul de date este Fruits360(Horea Muresan, Mihai Oltean, Fruit recognition from images using deep learning, Acta Univ. Sapientiae, Informatica Vol. 10, Issue 1, pp. 26-42, 2018.), continand peste 80 000 de poze din 120 de clase de fructe diferite.

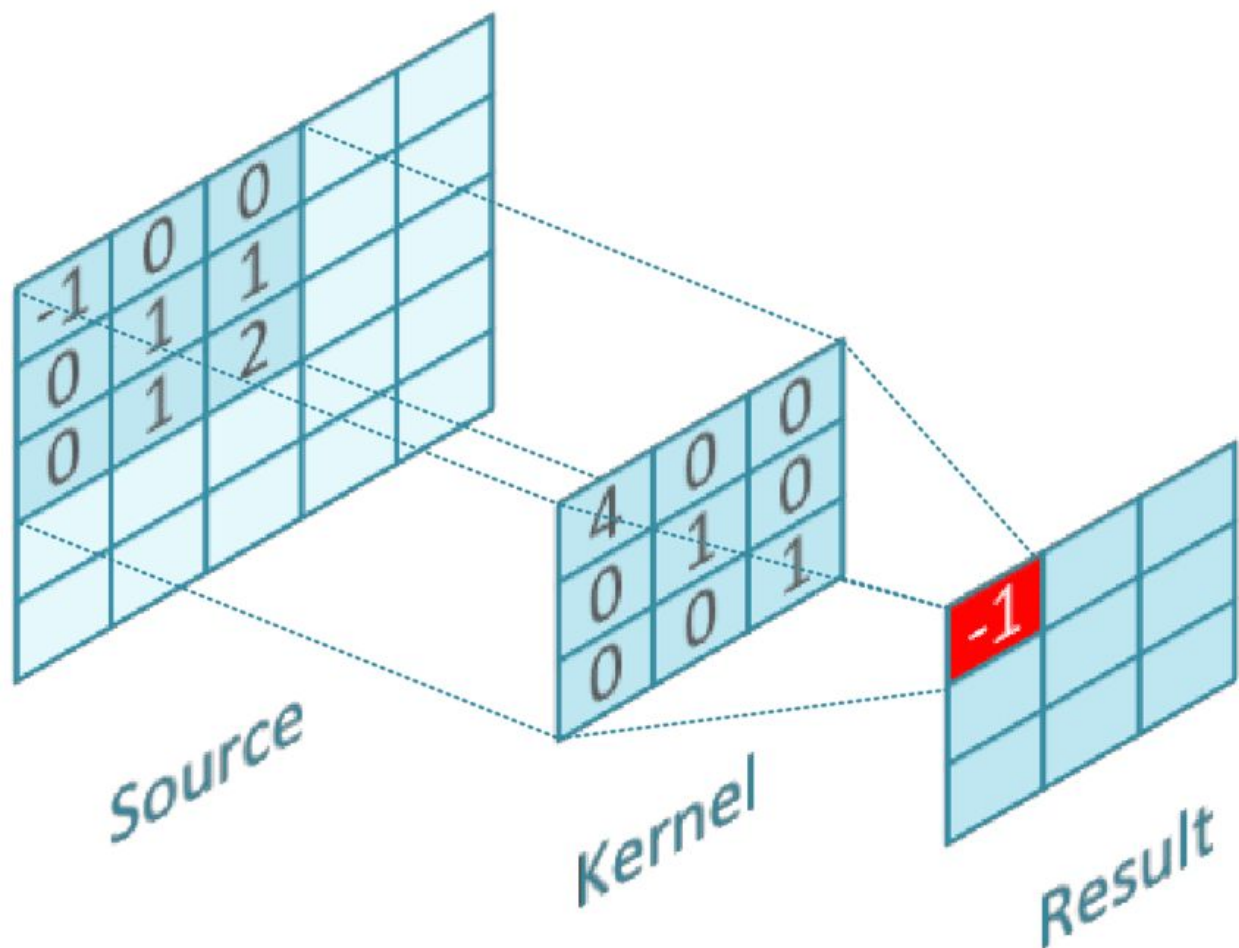
In designul retelei, am folosit urmatoarele straturi:

1. 16 noduri convolutionale, fiecare cu matricea convolutionala de (5,5)
2. Un strat de normalizare a datelor
3. Un strat de MaxPooling cu matricea de maxim de dimensiune (3,3), pentru reducerea dimensiunii datelor pe parcursul analizei datelor de intrare.
4. 32 noduri convolutionale, fiecare cu matricea convolutionala de (5,5)
5. Un strat de normalizare a datelor
6. Un strat de MaxPooling cu matricea de maxim de dimensiune (3,3), pentru reducerea dimensiunii datelor pe parcursul analizei datelor de intrare.
7. 64 noduri convolutionale, fiecare cu matricea convolutionala de (5,5)
8. Un strat de normalizare a datelor
9. Un strat de MaxPooling cu matricea de maxim de dimensiune (3,3), pentru reducerea dimensiunii datelor pe parcursul analizei datelor de intrare.
10. 128 noduri convolutionale, fiecare cu matricea convolutionala de (5,5)
11. Un strat de normalizare a datelor
12. Un strat de MaxPooling cu matricea de maxim de dimensiune (3,3), pentru reducerea dimensiunii datelor pe parcursul analizei datelor de intrare.
13. Un strat de aplatizare. Acest strat transforma tabloul multi-dimensional intr-un vector 2D
14. Un strat complet conectat la cel precedent, cu 1024 de componente
15. Un strat complet conectat la cel precedent, cu 256 de componente

16. Ultimul strat, cu 120 de clase, fiecare nod reprezentand probabilitatea ca poza de la datele de intrare sa fie respectiva clasa.

Ce este o matricea convolutionala ?

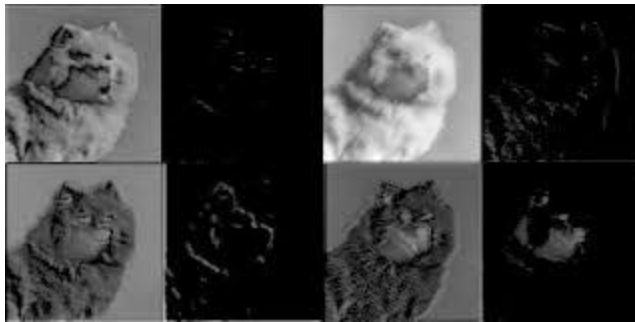
Este o matrice de o dimensiune data, care are anumite valori ce pot scoate in evidenta diferite trasaturi ale imaginii (linii verticale, linii orizontale, curbe etc) cand valorile intensitatii pixelilor imaginii sunt inmultite cu matricea.



In aceasta imagine, este simplificat procesul care se produce in cazurile reale.

Matricea cea mare(source) reprezinta in cazul nostru imaginea, Kernelul este matricea convolutionala si rezultatul este outputul stratului care contine nodul cu matrice convolutionala. Rezultatul este dat adunarea dintre produsul fiecarui element din Kernel cu corespondentul lui din matricea cea mare  $(-1 * 4 + 0 * 0 + 0 * 0 + 0 * 0 + 1 * 1 + 1 * 0 + 0 * 0 + 1 * 0 + 2 * 1 = -1)$ .

Acest caz este doar o simplificare a problemei reale. In cazul nostru, unde imaginea este formatata la dimensiunea de (100, 100) pixeli, matricea convolutionala se "plimba" pe imagine, outputul rezultand in alta matrice, denumita matrice de trasaturi(pentru ca filtrul, matricea convolutionala, scoate in evidenta diferite trasaturi ale imaginii, cum am explicat mai sus).

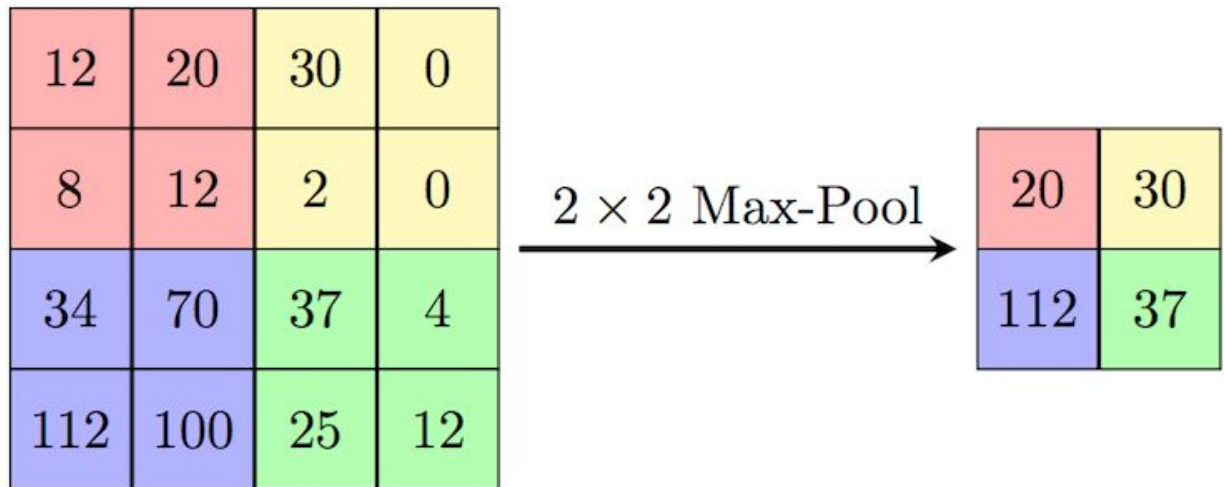


In imaginea de mai sus se poate observa extragerea trasaturilor unei imagini aplicand un strat convolutional.

Ce este o matrice de maxim?

O matrice de maxim este folosita in straturile de tip MaxPooling, si principiul de functionare este asemanator cu cel al matricei convolutionale. Matricea de maxim trece peste imagine in acelasi mod ca si matricea convolutionala, dar el alege de fiecare data valoarea maxima, formand cu ele o

noua matrice.

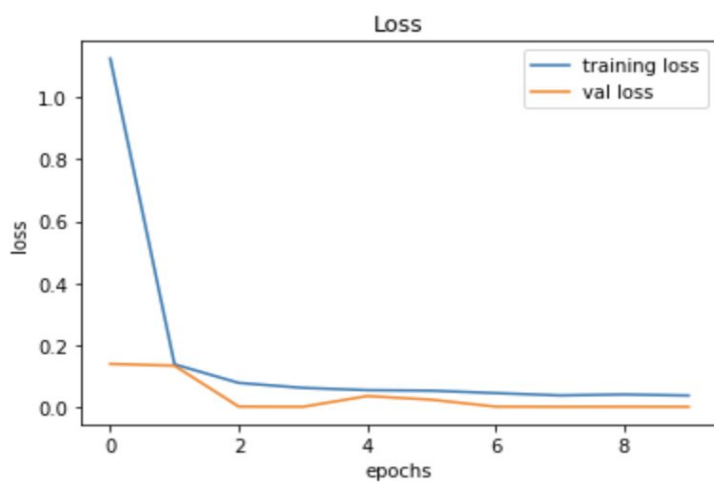
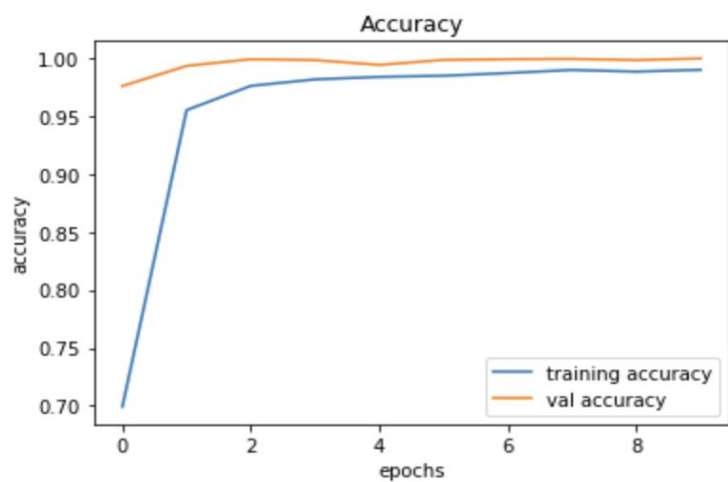


#### 4. Antrenarea, validarea, si testarea retelei.

Setul de date este impartit in 3 sectiuni. Sectiunea pentru antrenarea retelei. La inceput, retea are niste valori aleatorii la parametrii functiilor matematice din fiecare nod, dand niste rezultate evident, eronate. A doua sectiune este cea de validare, care verifica de-a lungul antrenarii verifica acuratetea retelei, iar a treia sectiune este cea de test, care verifica acuratetea retelei cand este gata antrenamentul si parametrii sunt setati.

Antrenarea se produce introducand cate o poza si calculand eroarea(loss), facand diferenta intre outputul produs de retea si cel corect. Dupa aceasta, algoritmul de backpropagation modifica parametrii in asa fel incat ca eroarea retelei sa fie cat mai mica(se creeaza un model matematic si se calculeaza minimul local). Acest proces de antrenare se realizeaza de mai multe ori peste sectiunea de antrenament a setului de date(in cazul nostru, 10 ori).





In graficele de mai sus este reprezentata acuratetea(primul grafic) pe setul de date de antrenament si cel de validare(albastru, respectiv portocaliu) de-a lungul a mai multor treceri prin setul de date(epochs), iar in graficul de jos este reprezentata eroarea rețelei.

## 5.Probleme intampinate. Imbunatatiri viitoare

Fiind o retea aflata inca nefinalizat integral, am intampinat unele probleme in anumite situatii. De exemplu, intampin erori cand in poza se afla mai multe fructe, sau cand fundalul pozei nu este alb. Pe viitor, ar trebui sa implementez un algoritm de detectare a obiectelor individuale intr-o poza(YOLO), si un algoritm de eliminare a backgroundului(Flood Fill de exemplu).

## 6.Bibliografie

<https://towardsdatascience.com/wtf-is-image-classification-8e78a8235acb>

<https://www.analyticsvidhya.com/blog/2020/02/learn-image-classification-cnn-convolutional-neural-networks-3-datasets/>

<https://towardsdatascience.com/my-tryst-with-deep-learning-german-traffic-data-set-with-keras-87970dfb18b7>

<https://www.kaggle.com/learn/deep-learning>

<https://www.deeplearningbook.org/>