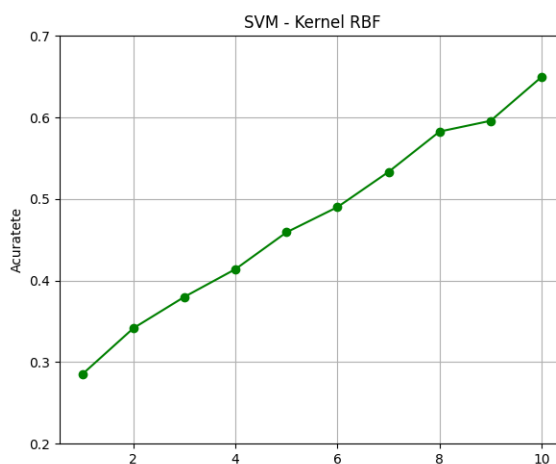
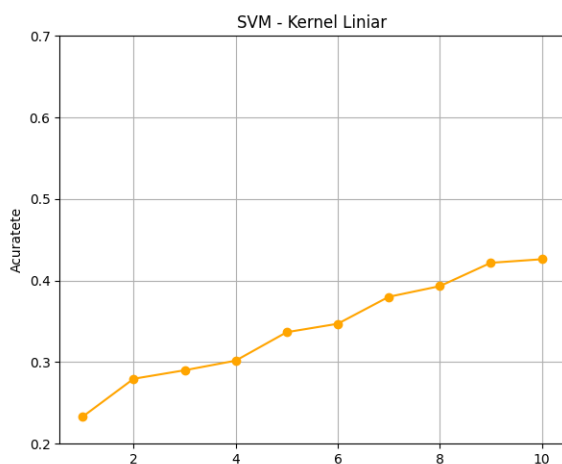


# Deepfake Classification Challenge

## Abordari implementate

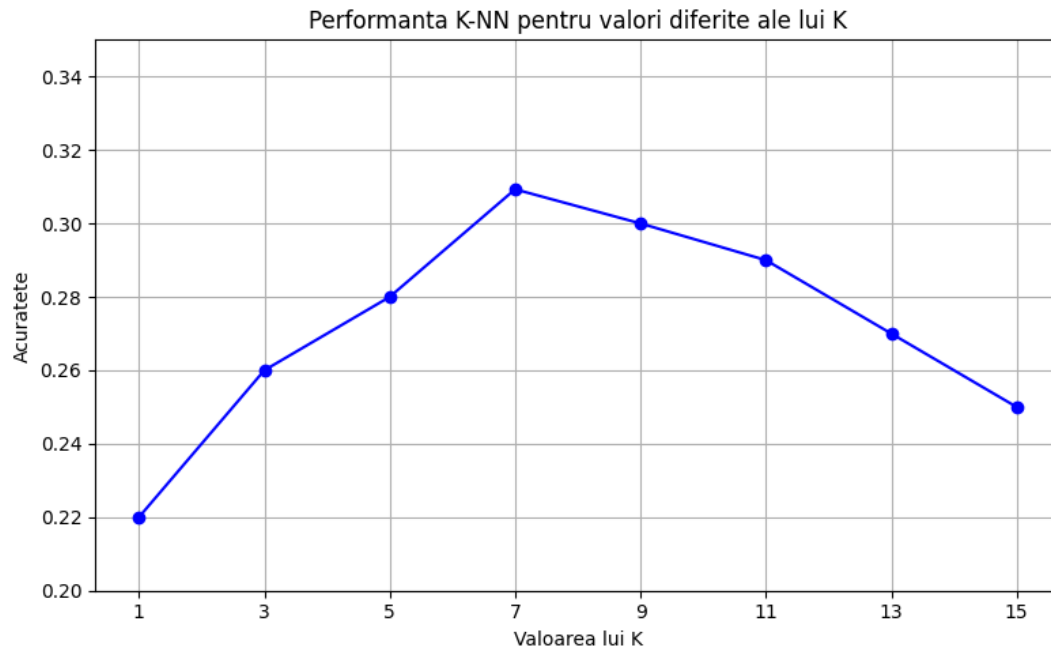
### 1. Suport Vector Machine (SVM)

- Au fost testate atat kernel-uri liniare, cat si RBF.
- Kernel liniar: acuratete de 43.33%.
- Kernel RBF cu tuning de hiperparametri: acuratete de 64%.



**2. K-Nearest Neighbors (K-NN)**- S-au testat mai multe valori ale lui K, folosind distanta Euclidiana.

- Cea mai buna acuratete obtinuta: 30.93%.

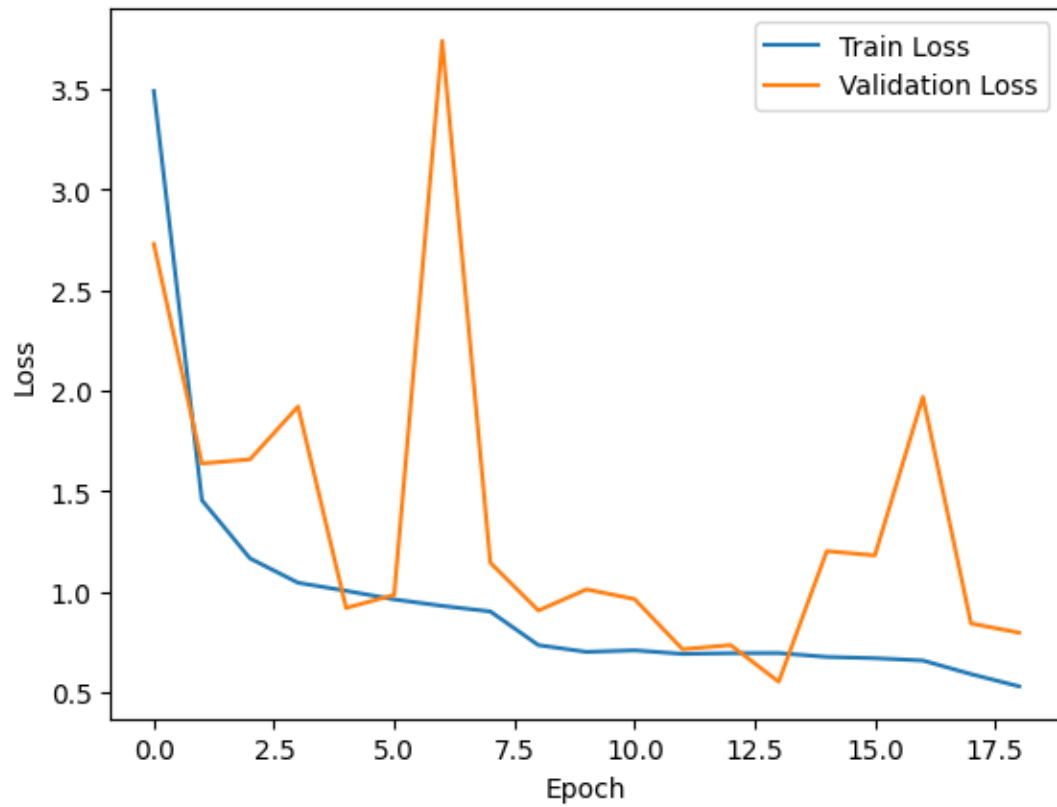


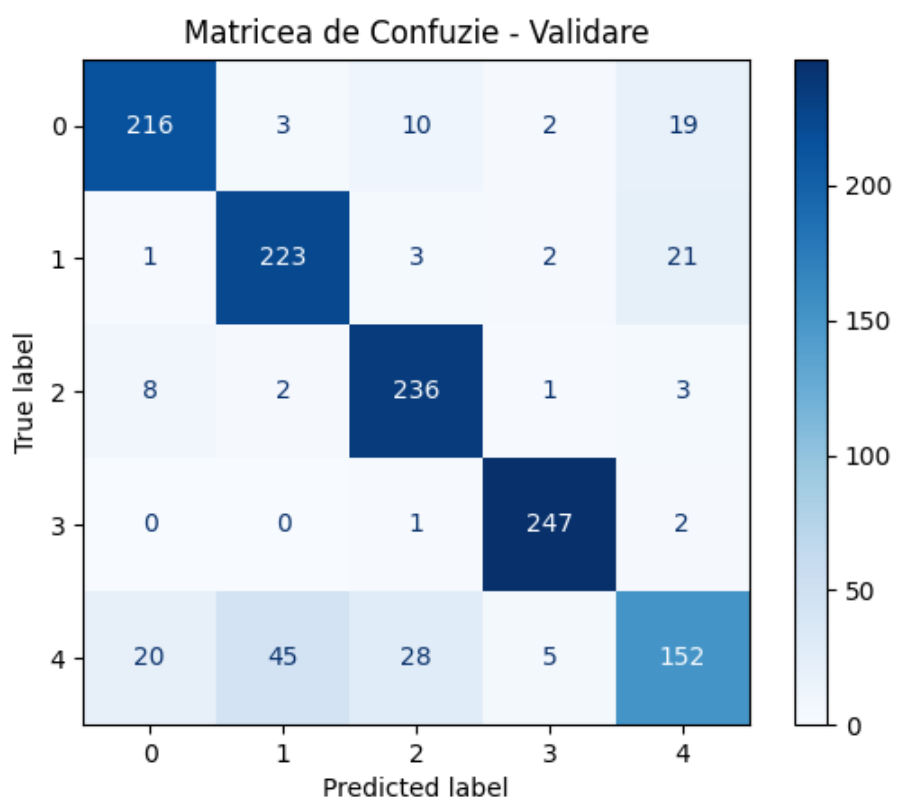
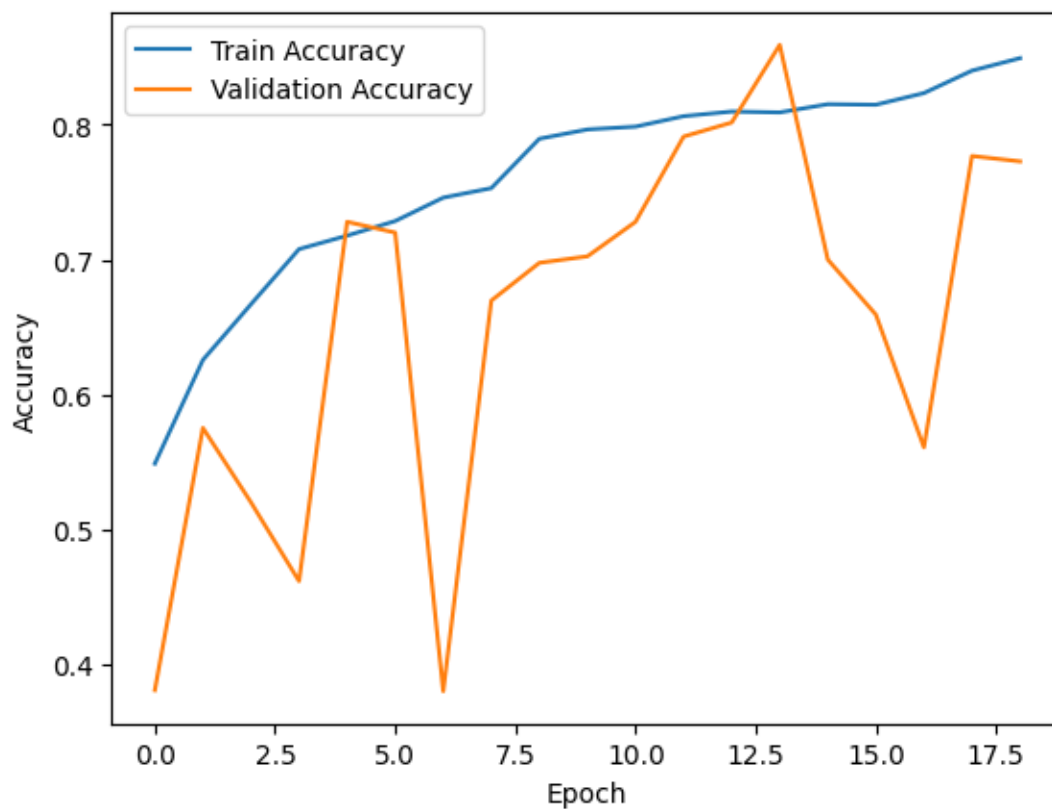
### 3. Convolutional Neural Networks (CNN)

- Un model de baza CNN a atins o acuratete de 80.2%.

- Modelul CNN optimizat, cu augmentare si arhitectura imbunatatita, a obtinut o acuratete de 85.4%.

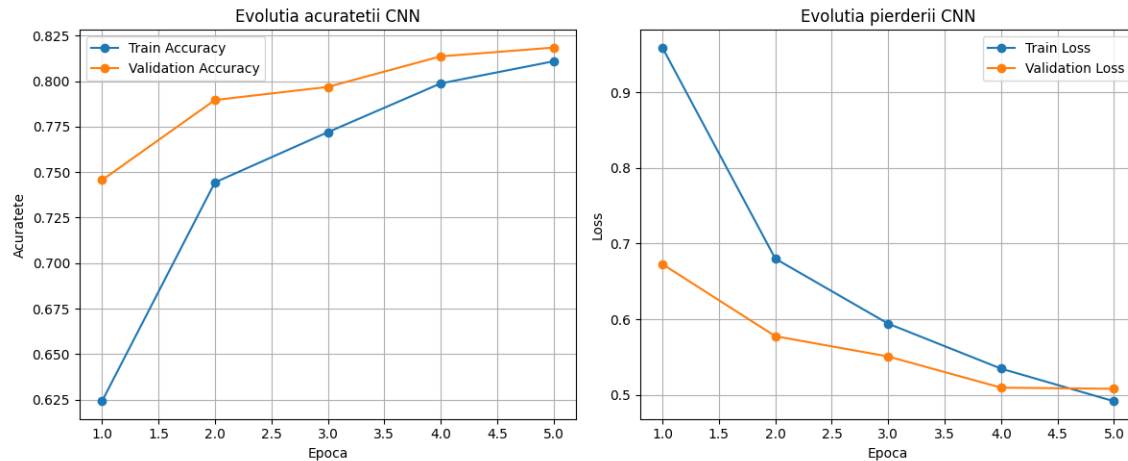
- Au fost incluse grafice pentru acuratete si pierdere in timpul antrenarii.





#### 4. Fine-tuning pe un model preantrenat (doar pentru curiozitate, nu a fost valid pentru concurs)

- A fost utilizat un model de tip deep learning preantrenat (MobileNetV2), care a fost ajustat pe setul nostru de date.
- Acest model a obținut o acurătate de 80.13% pe setul de validare.



#### Preprocesarea datelor si extragerea caracteristicilor

**Rescalare:** imaginile au fost normalizate cu valori între 0 și 1 (rescale=1./255).

**Augmentare (pentru datele de antrenament):**

- rotation\_range=20: rotiri aleatorii ale imaginii cu până la 20 de grade.
- width\_shift\_range=0.2, height\_shift\_range=0.2: translatarea imaginii pe orizontală și verticală.
- shear\_range=0.2: aplicare de efect de forfecare.
- zoom\_range=0.2: zoom aleatoriu.
- horizontal\_flip=True: întoarcerea imaginii pe orizontală.
- fill\_mode='nearest': completarea pixelilor lipsă.

#### Ajustarea hiperparametrilor

- Pentru SVM: s-a realizat cautare in grid pentru parametrii C si gamma (kernel RBF).
- Pentru CNN: s-au ajustat rata de invatare, batch size, numarul de straturi, rata de dropout si parametrii de augmentare.

### Tabel comparativ:

Model	Hiperparametrii	Acuratete
SVM (Linear)	Implicit	43.33%
SVM (RBF)	C = 10, Gamma = 0.01	64%
CNN Basic	LR = 0.001, Batch Size 32	80.02%
CNN Optimizat	LR = 0.0001, Dropout = 0.3, Augmentare	85.4%

### Arhitectura CNN strat cu strat

#### Conv2D + BatchNormalization + MaxPooling

##### 1. Conv2D(32)

- Aplica 32 de filtre (dimensiune 3x3) pe imagine.
- Detecteaza trasaturi simple: margini, colturi, texturi.
- Output: (98, 98, 32) – imaginea este usor micșorata.

##### 2. BatchNormalization()

- Normalizeaza activarile pentru a accelera antrenamentul și a stabili rețeaua.

##### 3. MaxPooling2D(2x2)

- Reduce dimensiunea spațială la jumătate: (49, 49, 32)
- Ajuta la reducerea complexității și la extragerea trasaturilor esențiale.

#### Flatten()

- Transforma tensorul (4, 4, 256) într-un vector de 4096 elemente.
- Pregătește datele pentru clasificare.

Dense(256)

- Strat complet conectat
- Activeaza o retea neuronală clasică pe trasaturile extrase.
- Are regularizare L2 și Dropout(0.5) pentru a preveni overfitting.

Dense(5)

- Strat final, cu 5 neuroni (pentru 5 clase).
- Activare softmax – ofera probabilitati pentru fiecare clasa.

### Rezultate

- Modelele CNN au avut performante superioare comparativ cu modelele clasice.
- Modelul CNN optimizat a oferit cele mai bune rezultate generale.
- Matricile de confuzie au aratat o reducere semnificativa a greselilor de clasificare pentru modelele CNN.