

Tema 2 ML

Oana Maria Bacaran, 342C5

Pe datasetul fashion:

3.1 Multi-Layer Perceptron (MLP) pe attributele extrase in etapa 1

Arhitectura rețelei

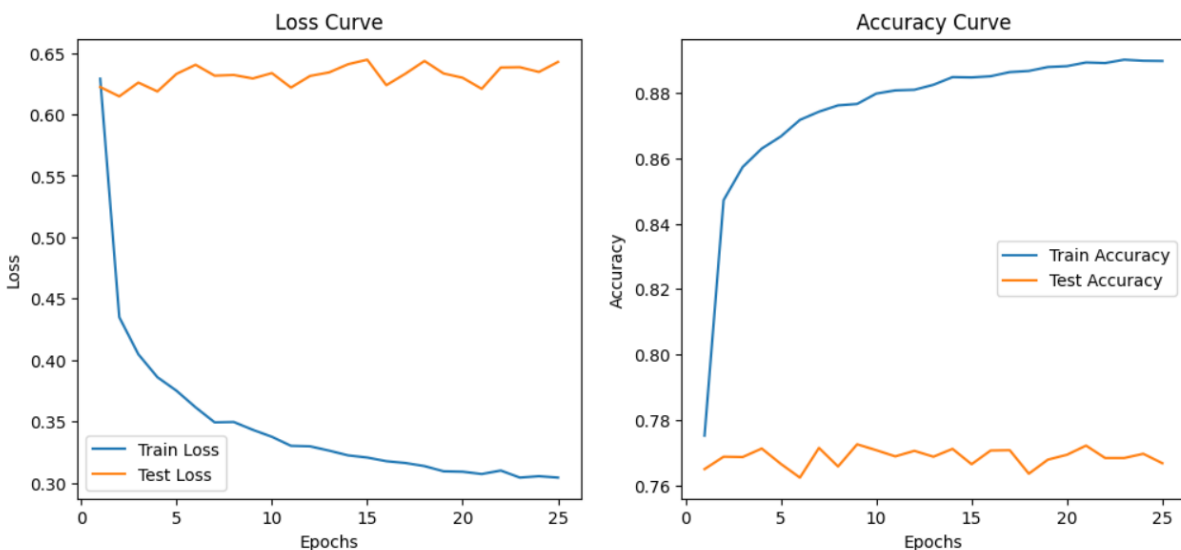
- Număr de straturi: 4 straturi dense (fully connected)
- Stratul de intrare cu 50 de neuroni (corespunzător celor 50 de attribute selectate).
- Două straturi ascunse: 256 de neuroni.
- Stratul de ieșire cu 10 neuroni (corespunzători claselor din Fashion-MNIST).
- Funcții de activare: ReLU pentru straturile ascunse și softmax pentru stratul de ieșire.

Caracteristicile procesului de antrenare

- Număr de epoci: 25
- Dimensiunea batch-ului: 64
- Funcția de pierdere: Categorical Cross-Entropy (`CrossEntropyLoss()`)
- Optimizator: Adam cu rata de învățare de 0.001 și weight decay de $1e-4$. Adaptive Moment Estimation este o tehnică de optimizare pentru gradient descent. Această metodă este foarte eficientă atunci când se lucrează cu probleme mari care implică multe date sau parametri. Necesită mai puțină memorie și este mai eficientă.
- Regularizare: Dropout (0.5) aplicat pe toate straturile pentru a preveni overfitting.

Rezultate

Comparativ cu cel mai bun model de la tema 1, Random Forest, acuratetea este puțin mai mică (cu 2%). Comparativ cu celelalte modele folosite la tema 1, rezultatele sunt la fel de bune sau chiar mai bune. Motivul probabil este folosirea regularizării (dropout și weight), care împiedică test loss-ul să crească în loc să scadă. Deoarece folosim dropout, test accuracy nu crește foarte mult, este destul de constantă, totuși loss-ul devine și el constant, deși fără utilizarea dropout avea tendința de a crește.



3.2 Multi-Layer Perceptron (MLP) pe imagini liniarizate

Arhitectura rețelei

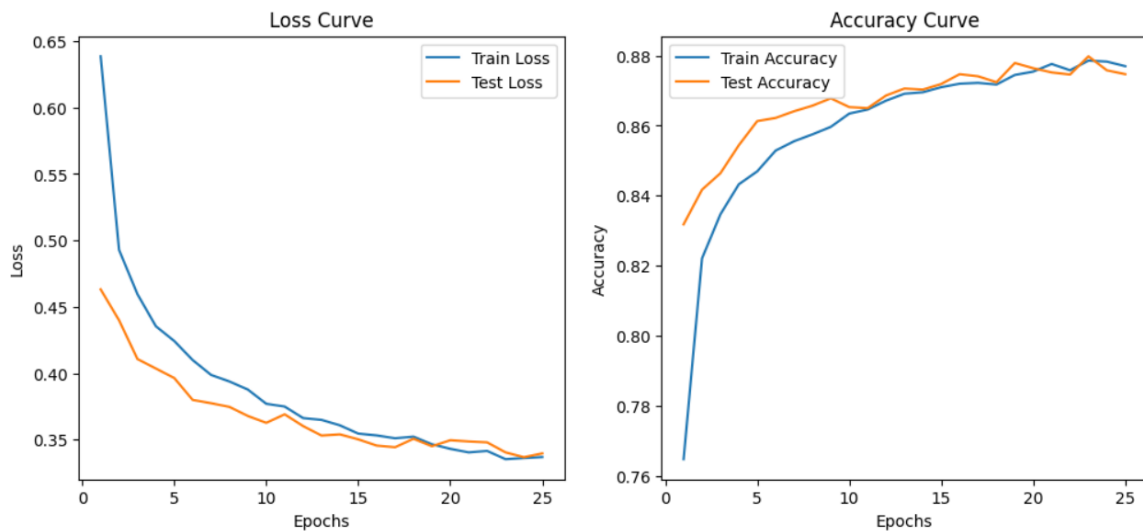
- Număr de straturi: 4 straturi dense
- Stratul de intrare cu 784 de neuroni (corespunzători pixelilor liniarizați ai imaginilor).
- Două straturi ascunse: 256 de neuroni.
- Stratul de ieșire cu 10 neuroni.
- Funcții de activare: ReLU pentru straturile ascunse și softmax pentru stratul de ieșire.

Caracteristicile procesului de antrenare

- Număr de epoci: 25
- Dimensiunea batch-ului: 64
- Funcția de pierdere: Categorical Cross-Entropy
- Optimizator: Adam cu rata de învățare de 0.001
- Regularizare: Dropout aplicat pe toate straturile ascunse (0.3-0.5).

Rezultate

Liniarizarea imaginilor permite modelului să capteze mai multe informații, dar creșterea dimensiunii vectorului de intrare adaugă complexitate. Acest model are o acurătate mai bună comparativ cu mlp-ul cu attribute extrase și are tendința mare de a scădea loss-ul ceea ce este de dorit.



3.3 Arhitectura de tip convoluțional:

Convoluții: Doua straturi convoluționale fiecare cu:

- Kernel de dimensiune 3x3.
- Normalizare prin lot (BatchNorm2d).
- Funcția de activare ReLU.
- Max pooling pentru reducerea dimensiunii.

Pooling adaptiv global: Reduce dimensiunea finală a caracteristicilor la o matrice de dimensiune 1x1.

Clasificator: Fully connected layer pentru a produce 10 clase.

Augmentari: Random Horizontal Flip, rotații și cropping utilizate pentru creșterea variabilității datelor de antrenare.

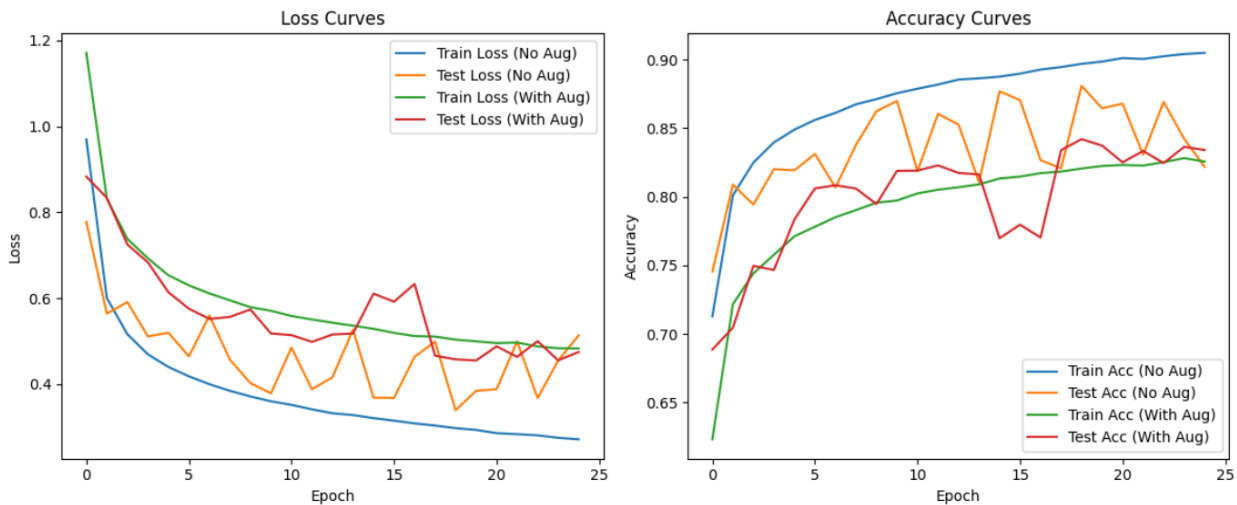
Antrenarea modelelor:

- Folosind CrossEntropyLoss pentru calculul erorii.
- Optimizator: Adam cu o rată de învățare de 0.001.
- Epoci: 25

Rezultate

Modelul fără augmentare converge mai rapid, dar augmentarea contribuie la generalizare mai bună pentru un set de testare complex.

Augmentarea introduce variabilitate în antrenare, ceea ce poate explica rata de învățare mai lentă la început.

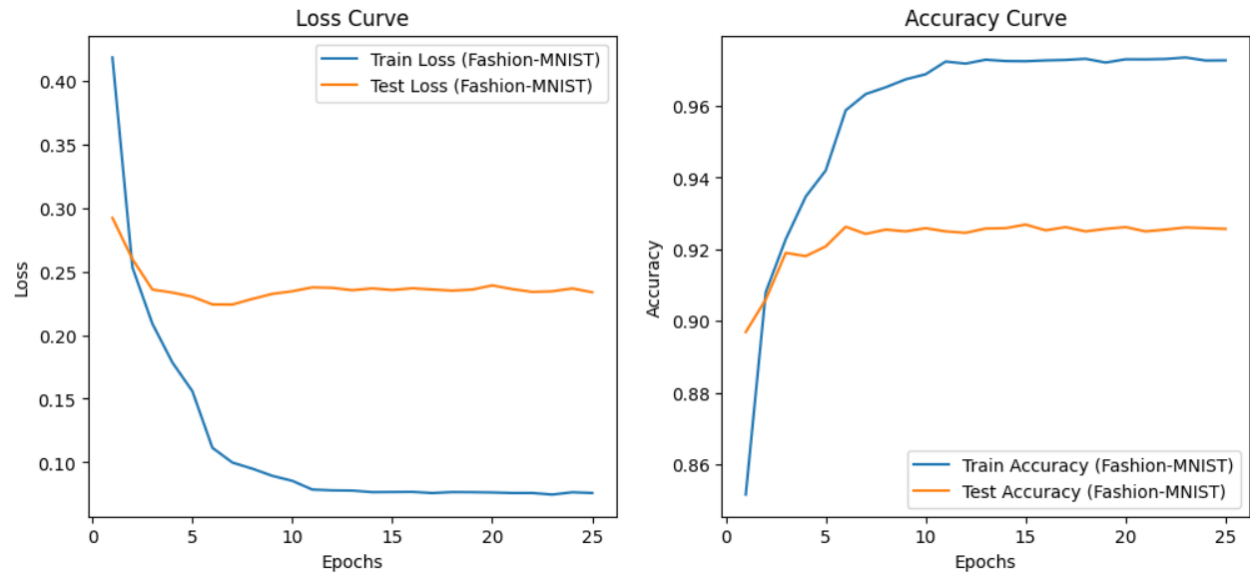


3.4 Utilizarea unei proceduri de finetuning peste arhitectura ResNet-18

Pentru antrenare folosim:

- Folosind CrossEntropyLoss pentru calculul erorii.
- Optimizator: SGD cu o rată de învățare de 0.001 si momentum 0.9
- Epoci: 25

Aceasta metoda are cele mai mari si constante rezultate deoarece folosim un model existent, care are cunoștințe preexistente capturate din seturi de date mari , dar este si cel care dureaza cel mai mult datorita marimii sale.



Tabel:

Model	Accuracy	Precision	Recall	F1-Score	Training Speed	Training Time
Tema 1 – Random Forest	0.78800	0.8343	0.8375	0.8371	-	-
3.1. MLP pe atributele extrase în etapa 1	0.7667	0.7612	0.7667	0.7606	22736.95 samples/sec	65.97 sec
3.2. Arhitectura de tip MLP direct peste imagini	0.8747	0.8750	0.8747	0.8742	3386.75 samples/sec	442.90 sec
3.3 Arhitectura de tip convoluțional neaugmentat	0.8218	0.8661	0.8218	0.8180	3521.18 samples/sec	425.99 sec
3.3 Arhitectura de tip convoluțional augmentat	0.8341	0.8382	0.8341	0.8284	1695.87 samples/sec	884.50 sec
3.4 Utilizarea unei proceduri de finetuning peste	0.9257	0.9255	0.9257	0.9256	1541.20 samples/sec	973.26 sec

arhitectura ResNet-18

MLP-ul aplicat pe imagini brute depășește CNN-ul neaugmentat, ceea ce indică faptul că arhitectura CNN simplă nu a fost suficient de profundă sau de bine configurată.

Prin adăugarea augmentării, CNN-ul devine mai performant, dar la un cost computațional ridicat.

Cel mai performant din punct de vedere al acuratetii a fost resnet.

Din punct de vedere al vitezei, MLP cu attribute deja extrase este cel mai rapid, dar este și cel mai slab calitativ. MLP cu imagini brute și DeepConvNet au viteze medii, în funcție de complexitatea modelului și variabilitatea augmentărilor, iar Resnet este cel mai lent.

Pentru Fruits:

3.1 Multi-Layer Perceptron (MLP) pe attributele extrase în etapa 1

Arhitectura rețelei

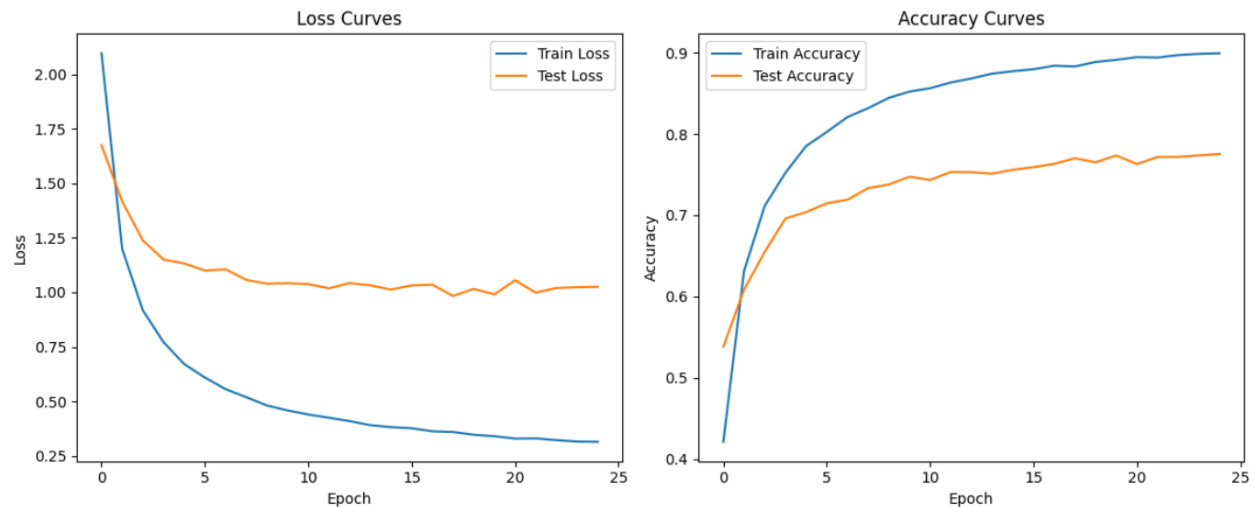
- Număr de straturi: 4 straturi dense (fully connected)
- Stratul de intrare cu 122 de neuroni (corespunzător celor 122 de attribute selectate).
- Două straturi ascunse: 128 de neuroni.
- Stratul de ieșire cu 70 neuroni (corespunzători numărului claselor din Fruits).
- Funcții de activare: ReLU pentru straturile ascunse și softmax pentru stratul de ieșire.

Caracteristicile procesului de antrenare

- Număr de epoci: 25
- Dimensiunea batch-ului: 64
- Funcția de pierdere: Categorical Cross-Entropy (CrossEntropyLoss())
- Optimizator: Adam cu rata de învățare de 0.001 și weight decay de $1e-4$
- Regularizare: Dropout (0.5) aplicat pe toate straturile pentru a preveni overfitting.

Rezultate

Comparativ cu cel mai bun model de la tema 1. Logistic Regression, acuratetea este puțin mai mică (cu 7%). Motivul probabil este folosirea regularizării (dropout și weight), care împiedică test loss-ul să crească în loc să scadă, de asemenea scăderea numărului de attribute cu select percentile de 9% de la 1300 la 122.



3.2 Multi-Layer Perceptron (MLP) pe imagini liniarizate

Arhitectura rețelei

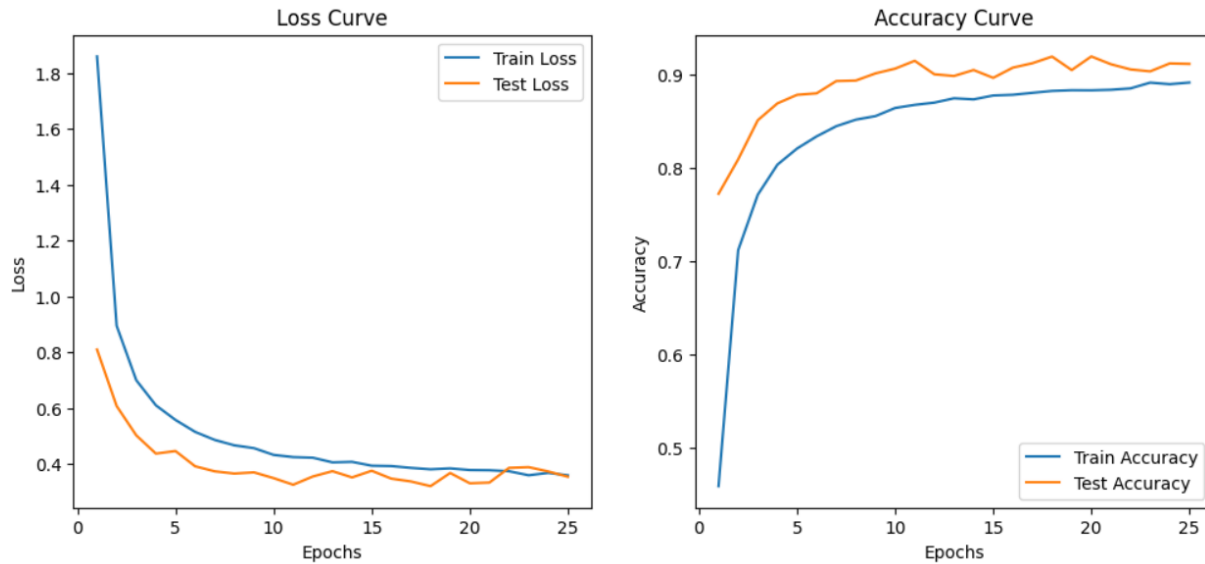
- Număr de straturi: 4 straturi dense
- Stratul de intrare cu 3x32x32 de neuroni (corespunzători pixelilor liniarizați ai imaginilor, am micșorat imaginile pentru un timp de rulare mai mic)
- Două straturi ascunse: 256 de neuroni.
- Stratul de ieșire cu 141 neuroni, numărul de clase din fruits
- Funcții de activare: ReLU pentru straturile ascunse și softmax pentru stratul de ieșire.

Caracteristicile procesului de antrenare

- Număr de epoci: 25
- Dimensiunea batch-ului: 64
- Funcția de pierdere: Categorical Cross-Entropy
- Optimizator: Adam cu rata de învățare de 0.001
- Regularizare: Dropout aplicat pe toate straturile ascunse (0.3-0.5).

Rezultate

Liniarizarea imaginilor permite modelului să capteze mai multe informații, dar creșterea dimensiunii vectorului de intrare adaugă complexitate. A fost nevoie de aplicarea regularizării dropout la fiecare strat pentru a împiedica overfit-ul și creșterea test loss-ului.



3.3 Arhitectura de tip convoluțional:

Convoluții: Trei straturi convoluționale adaptate la 3 canale de culori fiecare cu:

- Kernel de dimensiune 3x3.
- Normalizare prin lot (BatchNorm2d).
- Funcția de activare ReLU.
- Max pooling pentru reducerea dimensiunii.

Pooling adaptiv global: Reduce dimensiunea finală a caracteristicilor la o matrice de dimensiune 1x1.

Clasificator: Fully connected layer pentru a produce 10 clase.

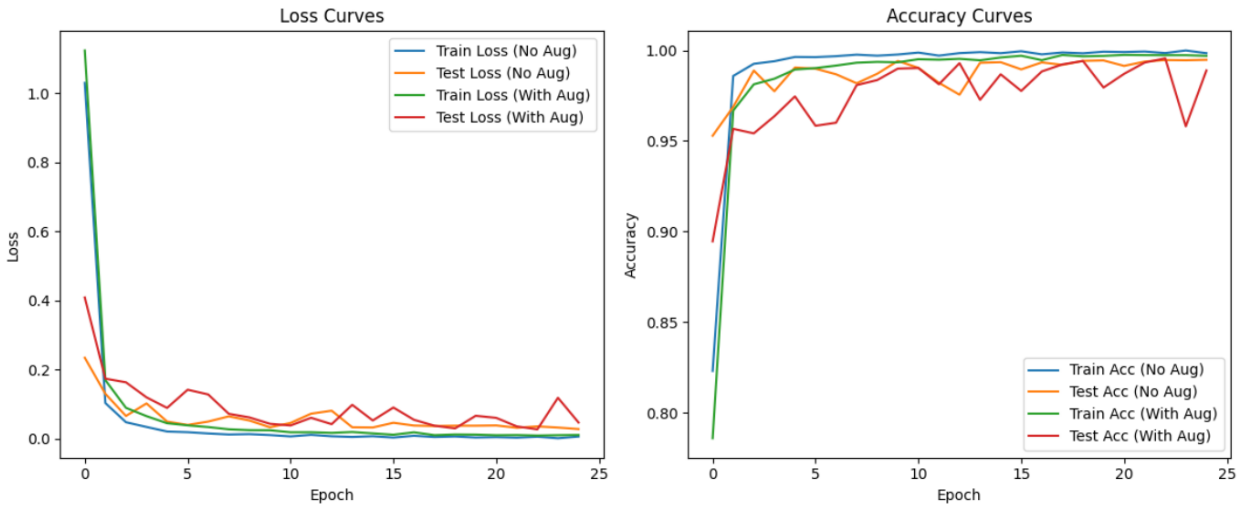
Augmentari: Random Horizontal Flip, rotații și cropping utilizate pentru creșterea variabilității datelor de antrenare.

Antrenarea modelelor:

- Folosind CrossEntropyLoss pentru calculul erorii.
- Optimizator: Adam cu o rată de învățare de 0.001.
- Epoci: 25

Modelul fără augmentare aduce cele mai bune rezultate, ceea ce înseamnă că numărul de straturi convoluționale este cel potrivit.

Augmentarea introduce variabilitate în antrenare, ceea ce poate explica rata de învățare mai lentă la început.

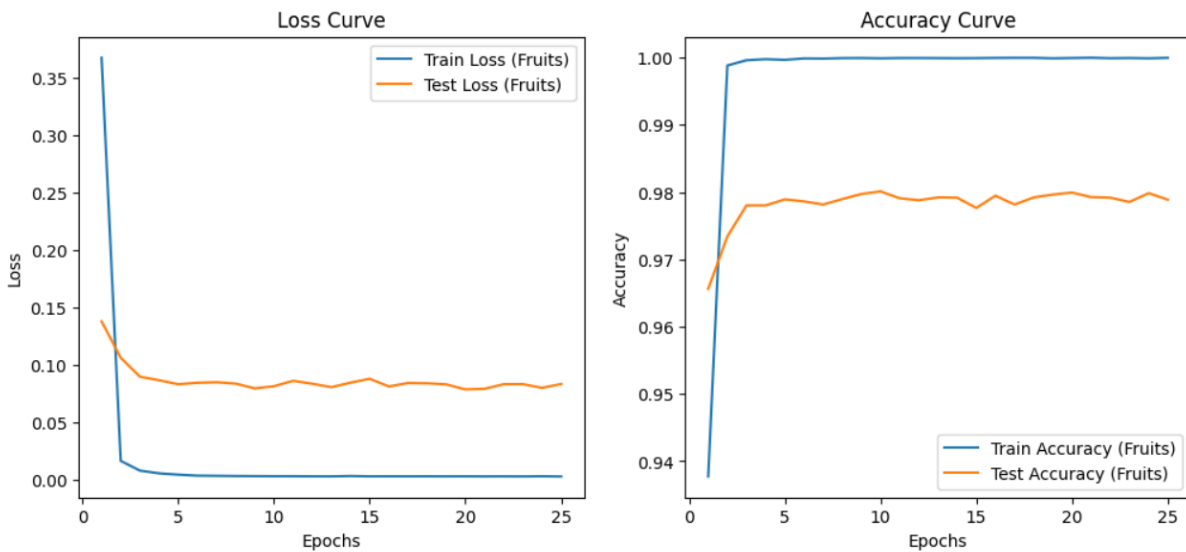


3.4 Utilizarea unei proceduri de finetuning peste arhitectura ResNet-18

Pentru antrenare folosim:

- Folosind CrossEntropyLoss pentru calculul erorii.
- Optimizator: SGD cu o rată de învățare de 0.001 si momentum 0.9
- Epoci: 25

Aceasta metoda are printre cele mai mari si constant rezultate deoarece folosim un model existent, care are cunoștințe preexistente capturate din seturi de date mari. Acuratetea se apropie de 100% in ambele cazuri train si test, iar loss-ul este mic ceea ce este bine deoarece pierdem putine date initiale despre imagini.



Tabel:

Model	Accuracy	Precision	Recall	F1-Score	Training Speed	Training Time
Tema 1 – Logistic Regression	0.85100	0.82800	0.79300	0.81	-	-
3.1. MLP pe attributele extrase în etapa 1	0.7754	0.7788	0.7754	0.7695	21988.82 samples/sec	80.14 sec
3.2. Arhitectura de tip MLP direct peste imagini	0.9110	0.9190	0.9110	0.9081	1504.26 samples/sec	1171.52 sec
3.3 Arhitectura de tip convoluțional neaugmentat	0.9905	0.9920	0.9905	0.9902	1250 samples/sec	1h 30min 31sec
3.3 Arhitectura de tip convoluțional augmentat	0.9746	0.9790	0.9746	0.9743	806.32 samples/sec	
3.4 Utilizarea unei proceduri de finetuning peste arhitectura ResNet-18	0.9789	0.9809	0.9789	0.9782	1217.36 samples/sec	1447.62 sec

CNN-ul este cel mai lent deoarece foloseste imaginile la scala lor originala, dar are si rezultatele cele mai bune deoarece sunt pastrate cele mai multe informatii despre imagini. Cel mai rapid este, la fel ca la fashion, MLP cu attribute extrase, dar are si cele mai proaste rezultate deoarece are putine attribute a date de intrare.