Université Perpignan Via Domitia

Optimisation IA Embarquée

Étude de compromis Performance/Ressources sur Fashion MNIST DATASET

Majid GHORBANNEZHAD

Année Universitaire 2025 - 2026

Introduction

Problématique

- Comment concilier performance IA et ressources limitées des systèmes embarqués ?

Objectif

- Identifier l'architecture optimale pour la classification Fashion-MNIST sur matériel contraint.

Approche : Comparaison systématique

- Autoencodeurs (CNN vs MLP) + Classifieurs (CNN vs MLP)
- Métriques : Précision / Vitesse / Consommation

Enjeu

- Trouver le meilleur compromis pour déploiement réel sur systèmes frugaux.

1. Méthodologie et Architecture

1.1. Datasets :

Fashion-MNIST

- Images 28×28 normalisées.
- 70 000 images (10 classes).
- Images en niveaux de gris.

Figure 1. Fashion-MNIST Datasets

1.2. Architecture Autoencodeur CNN

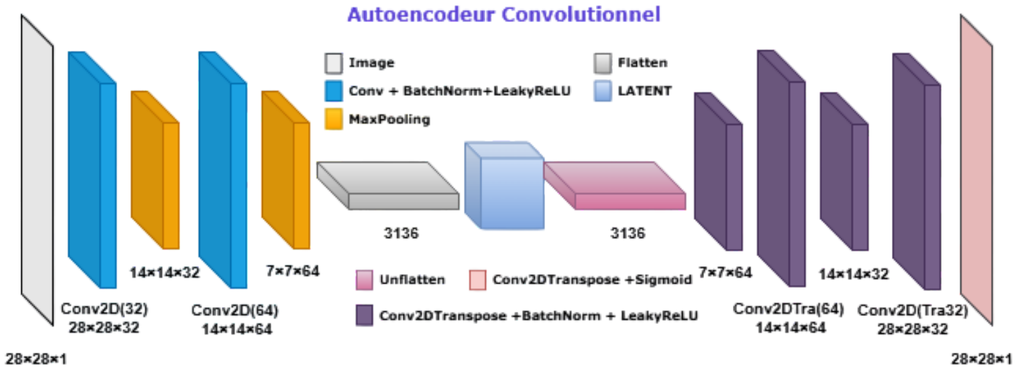


Figure 2. Structure de AE CNN

1.3. Architecture du réseau CNN

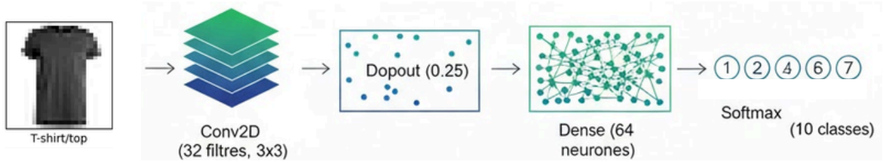


Figure 3. Structure du modèle de classification

1.4. Métriques de Reconstruction

- **MSE :** Erreur pixel-à-pixel (plus petit = meilleur).
- **PSNR :** Fidélité en dB.
- **SSIM :** Similarité structurelle (proche de 1 = reconstruction quasi parfaite).

1.5. Protocoles de Classification

Trois scénarios sont testés :

1. **Cas 1 :** Entraînement + test sur images originales.
2. **Cas 2 :** Entraînement + test sur images reconstruites par l'AE.
3. **Cas 3 :** Entraînement sur un mélange d'images originales et reconstruites pour évaluer la robustesse.(ratio 1:1)

1.6. Infos système et outils

- **Langage utilisé :** Python
- **Plateformes :** Google Colab (cloud) et Spyder (local)
- Machine locale :
- **Processeur :** AMD Ryzen 3 3200U @ 2.60 GHz
- **RAM installée :** 8 Go (5.92 Go utilisables)
- **Système :** Windows 64 bits, architecture x64

2. Étude Expérimentale de base : AE CNN + Classification CNN

2.1. Impact de la taille de l'espace latent

Afin de réaliser cette expérimentatin, nous faisons un test pour 6 dimensions différentes de l'espace latent:

Dimension (d)	MSE (↓)	PSNR (dB) (↑)	SSIM (↑)	Observation
20	0.01031	19.87	0.7615	Reconstruction floue
40	0.00723	21.41	0.8134	Gain de netteté significatif
60	0.00585	22.33	0.8440	Détails des textures visibles
80	0.00511	22.92	0.8625	Convergence de la qualité
100	0.00466	23.31	0.8726	Convergence de la qualité
128	0.00366	24.40	0.8988	Haute-fidélité

Tableau 1. Impact de dimension de l'espace latent

- **Rôle du latent :**
  - **Trop petit** → Perte de détails.
  - **Trop grand** → Compression moins utile.
- **Tendance globale :**
  - **Q**ualité ↑ quand la dimension ↑ (MSE ↓, PSNR/SSIM ↑).
- **Zones clés :**
  - 20 → 60 : Forte amélioration (netteté, textures).
  - ≥ 80 : Saturation des gains (qualité converge).
- **Choix optimal :**
  - **60 dimensions : meilleur compromis → SSIM élevé + MSE petite.**
  - Au-delà de cette valeur, l'augmentation de la complexité ne justifie plus les gains marginaux de SSIM et de MSE, tandis qu'en deçà, la perte d'informations texturales dégrade la classification.

2.2. Impact de l'espace latent sur la classification

Dimension latente	SSIM	Acc. Originale	Acc. Reconstruite	Acc. Combinée
40	0.8113	89.87 %	84.74 %	90.45 %
60	0.8297	90.02 %	85.20 %	90.88 %
80	0.8581	89.31 %	86.79 %	90.35 %
100	0.8602	89.61 %	87.96 %	90.20 %
128	0.8655	90.35 %	88.40 %	90.77 %

Tableau 2. Impact de dimension de l'espace latent sur la classification

Analyse des Matrices de confusion

- **Images originales :**
  - Confusions concentrées sur les classes proches (Shirt / T-shirt / Pullover).
- **Images reconstruites :**
  - Classes simples stables, mais Shirt très sensible à la compression
- **Données combinées :**
  - Robustesse accrue aux formes globales, mais risque de confusion accrue pour les classes complexes lorsque la reconstruction lisse trop les textures.

Conclusion

- Cette étude de sensibilité confirme que le choix de la dimension latente résulte d'un compromis entre compression, qualité de reconstruction et effet régularisant sur la classification.

3. Autoencodeur MLP vs CNN

3.1. Architecture Autoencodeur MLP

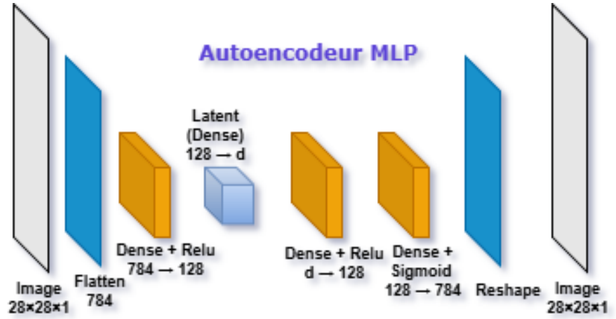


Figure 4. Structure de AE MLP

3.2. Analyse des résultats de reconstruction

Dimension d	MSE (MLP)	MSE (Conv)	SSIM (MLP)	SSIM (Conv)	Différence SSIM
40	0.0113	0.00723	0.7372	0.8113	−0.0741
60	0.0105	0.00585	0.7534	0.8442	−0.0908
80	0.0102	0.00511	0.7592	0.8581	−0.0989
100	0.0098	0.00466	0.7672	0.8602	−0.0930
128	0.0095	0.00366	0.7722	0.8655	−0.0933

Tableau 3. Comparaison d'architectures MLP vs CNN

3.3. Impact AE MLP sur la classification

Dimension latente d	Acc. Originale	Acc. Reconstruite	Acc. Combinée
40	89.90 %	85.80 %	90.25 %
60 (choix étudié)	88.04 %	85.63 %	89.45 %
80	90.38 %	86.88 %	90.15 %
100	90.00 %	86.67 %	90.25 %

Tableau 4. Synthèse des performances de classification (Architecture MLP)

- **AE MLP**
  - Perte des relations spatiales.
  - Reconstructions plus floues.
  - SSIM plus faible (≈ 0.75).
  - Détails fins et textures moins bien préservés.
- **AE Convolutif**
  - Modélisation spatiale beaucoup plus efficace.
  - Contours nets, textures mieux conservées.
  - SSIM nettement supérieur (≈ 0.84).
  - Reconstruction visuellement plus fidèle.
- **Impact AE MLP sur la classification**
  - Légèrement moins performant → perte de détails discriminants.
  - Confusions accrues pour les classes visuellement proches : Shirt / T-shirt / Pullover.
  - Classes simples stables : Trouser, Sandal.
  - Performances globales proches de celles de l'AE CNN, malgré les limites du MLP.

Conclusion

- Performances similaires sur données originales et combinées dans les deux autoencodeurs étudiés.
- Malgré une bonne fidélité de reconstruction, l'augmentation par AE ne génère pas de bénéfice significatif.

4. Analyse Temps/Performance : AE CNN + Classification CNN vs MLP

Dans cette étape, tout en conservant l'architecture CNN pour l'autoencodeur, nous examinons l'impact d'un classifieur MLP par rapport à un CNN en termes de précision et de temps de traitement.

4.1. Architecture du réseau MLP

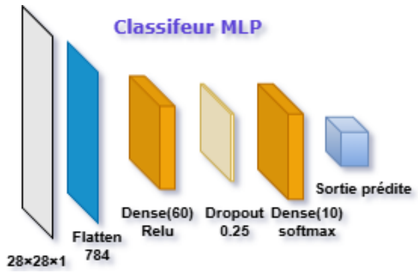


Figure 5. Structure de Classification MLP

4.2. MLP vs CNN : Temps de traitement

Dim. d	Config.	Ent. (CNN)	Ent. (Dense)	Éva. (CNN)	Éva. (Dense)	Gain Ent.	Gain Éva.
100	Originales	161.31 s	12.65 s	2.11 s	0.76 s	12.7x	2.8x
	Reconstruites	145.20 s	14.88 s	2.01 s	0.74 s	9.8x	2.7x
	Combinées	316.93 s	24.76 s	3.05 s	0.77 s	12.8x	4.0x
80	Originales	141.17 s	11.25 s	2.77 s	0.66 s	12.5x	4.2x
	Reconstruites	175.62 s	12.85 s	1.94 s	0.64 s	13.7x	3.0x
	Combinées	314.13 s	22.65 s	1.98 s	0.67 s	13.9x	3.0x
60	Originales	142.38 s	13.44 s	1.96 s	0.77 s	10.6x	2.5x
	Reconstruites	174.22 s	12.95 s	2.86 s	0.83 s	13.5x	3.4x
	Combinées	312.67 s	23.42 s	2.82 s	0.74 s	13.4x	3.8x
40	Originales	156.94 s	15.50 s	2.79 s	0.75 s	10.1x	3.7x
	Reconstruites	158.65 s	12.13 s	3.23 s	0.73 s	13.1x	4.4x
	Combinées	311.52 s	24.26 s	1.91 s	0.78 s	12.8x	2.4x

Tableau 5. Temps de traitement de classification MLP vs CNN

- **Gain d'Entraînement global MLP** ≈ 12.4x plus rapide que CNN en moyenne.
- **Gain d'Évaluation global MLP** ≈ 3.3x plus rapide que CNN en moyenne.

4.3. MLP vs CNN : Précision de classification

Dim. *d*	Acc. Orig. (CNN)	Acc. Orig. (MLP)	Acc. Rec. (CNN)	Acc. Rec. (MLP)	Acc. Comb. (CNN)	Acc. Comb. (MLP)
40	89.87 %	88.06 %	84.74 %	87.03 %	90.45 %	88.70 %
60	90.02 %	88.23 %	85.20 %	87.31 %	90.88 %	88.74 %
80	89.31 %	88.42 %	86.79 %	86.05 %	90.35 %	88.29 %
100	89.61 %	88.47 %	87.96 %	86.67 %	90.20 %	88.37 %

Tableau 6. Précision de classification de classification MLP vs CNN

- **Écart minimal :** Seulement 1-2% de différence.
- **Performances comparables :** 88-89% (MLP) vs 89-90% (CNN).

4.4. Décision de réseau de classification

Critère	CNN	Dense (MLP)	Recommandation
Précision	★★★★☆ (90%)	★★★☆☆ (88%)	CNN légèrement mieux
Vitesse Entraînement	★☆☆☆☆ (156s)	★★★★★ (13s)	MLP bien mieux
Vitesse Inférence	★★☆☆☆ (2.4s)	★★★★☆ (0.73s)	MLP bien mieux
Consommation Énergie	★★☆☆☆	★★★★☆	MLP mieux
Simplicité	★★☆☆☆	★★★★☆	MLP mieux
Score Total	**12/25**	**20/25**	☑ **MLP RECOMMANDÉ**

Tableau 7. Comparaison réseau CNN / MLP

- Le compromis performance/temps est excellent.
  - -1-2% de précision pour ×12 gain en vitesse.
- L'inférence plus rapide est critique en embarqué (temps réel, batterie).
- Moins de complexité = moins de bugs, maintenance plus simple.
- Adaptabilité : Facile à optimiser/quantifier pour matériel contraint.

5. Analyse Temps/Performance : AE MLP vs CNN + Classification MLP

Cette fois-ci, en conservant le MLP comme réseau de classification, nous nous intéressons au choix de l'autoencodeur le plus adapté à notre cas d'étude et compatible avec les contraintes d'un système embarqué.

5.1. AE MLP vs CNN : Temps de traitement et Précision de classification

Dim. d	Ent. AE MLP	Ent. AE CNN	Acc. Orig. CNN+MLP	Acc. Orig. CNN+MLP	Acc. Rec. MLP+MLP	Acc. Rec. CNN+MLP	Acc. Comb. MLP+MLP	Acc. Comb. CNN+MLP
40	69.27 s	3713.01 s	87.50%	88.06%	84.97%	87.03%	87.59%	88.70%
60	76.29 s	3585.09 s	87.08%	88.23%	85.32%	87.31%	87.66%	88.74%
80	73.31 s	3648.83 s	87.20%	88.42%	85.24%	86.05%	87.58%	88.29%
100	82.50 s	3628.69 s	87.22%	88.47%	85.68%	86.67%	87.95%	88.37%

Tableau 8. Comparaison Temps/Performance AE MLP vs CNN + Classification MLP

Performances (Précision)

- **Différence moyenne :** ~1 - 1.5% en faveur de AE CNN
- **Pertes acceptables :** -0.75% à -2% selon configuration

Gains Temporels

- Entraînement AE : 44 - 54× plus rapide (MLP)
- Reconstruction : 4× plus rapide (MLP)
- Classifieur : Déjà 12× plus rapide (MLP vs CNN)

- **Choix optimal pour Système Embarqué**

Un réseau de classification MLP sans Autoencodeur

```
model = tf.keras.Sequential([
    Flatten(input_shape=(H, W, C)),     # Aplatissement entrée image
    Dense(60, activation='relu'),       # Couche cachée compacte
    Dropout(0.25),                      # Régularisation contre l'overfitting
    Dense(10, activation='softmax')     # Couche sortie (10 classes)
])
```

- **MLP seul :** Plus rapide, plus simple, précision similaire.
- **AE inutile :** L'augmentation des données n'améliore pas la classification.
- **Gain direct :** Éviter la surcharge de l'AE (entraînement + reconstruction).

6. Impact du LeakyReLU sur la Reconstruction (AE CNN)

La fonction ReLU (Rectified Linear Unit) est définie comme ***f(x) = max(0,x)***. Elle est largement utilisée en raison de sa simplicité et de son efficacité à résoudre le problème de vanishing gradient, en activant uniquement les neurones ayant une entrée positive. Une variante fréquemment employée, Leaky ReLU, est définie comme ***f(x) = max(αx,x)*** où ***α*** est une petite pente positive pour les valeurs négatives. Cela permet d'éviter le problème des neurones morts en assurant un gradient non nul même pour les entrées négatives, améliorant ainsi la stabilité et les performances de l'apprentissage dans certains réseaux plus profonds ou sensibles à l'initialisation.

- **ReLU standard**
  - Convergence correcte mais plateau rapide
  - Reconstruction limitée avec perte de détails
  - Performances métriques inférieures.

- **LeakyReLU :**
  - Apprentissage plus profond et plus stable
  - Gain perceptuel net, textures mieux restituées
  - Amélioration significative sur les 3 métriques

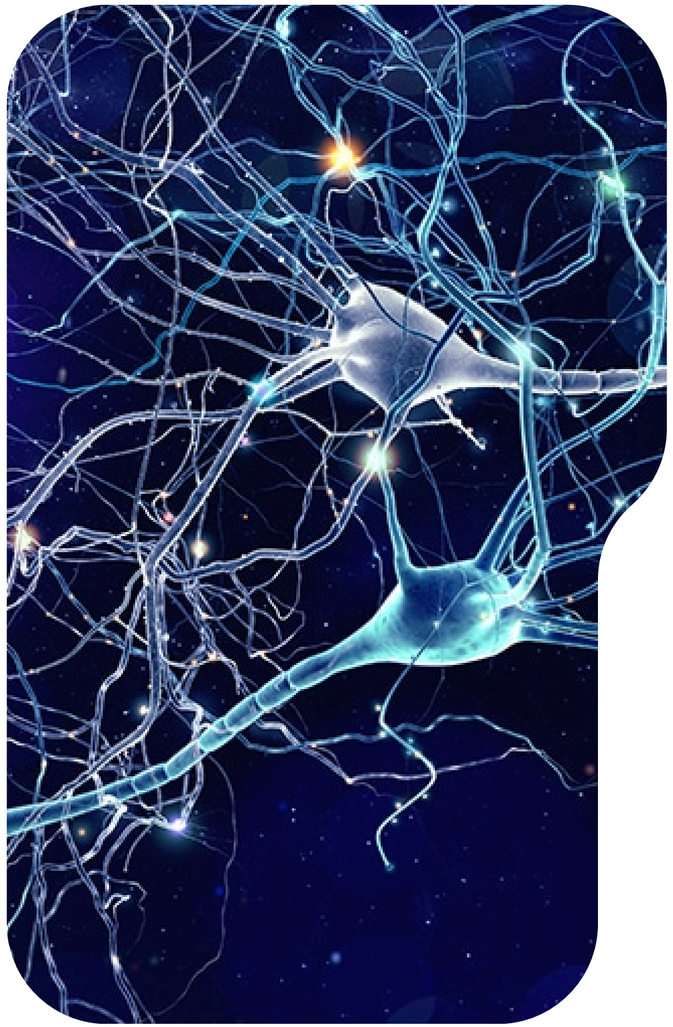
- **Pourquoi çela fonctionne ?**
  - LeakyReLU évite les neurones morts en gardant un gradient non nul pour les activations négatives

Métrique	ReLU Standard	LeakyReLU	Amélioration
MSE	0.0084	0.0058	↓ 31%
PSNR	20.76 dB	22.33 dB	+1.57 dB
SSIM	0.7629	0.8444	+0.0815

Tableau 9. Comparaison ReLU vs LeakyReLU

Conclusion

- Pour un latent modéré (60), LeakyReLU améliore fortement la qualité de reconstruction et la richesse de la représentation compressée.



Conclusion

Cette étude comparative approfondie sur le dataset Fashion-MNIST démontre que pour les systèmes embarqués aux ressources limitées, l'approche la plus optimale est l'élimination complète de l'autoencodeur au profit d'un classifieur MLP simple et direct.

1. L'AUTOENCODEUR N'APPORTE PAS DE BÉNÉFICE SIGNIFICATIF POUR LA CLASSIFICATION

- L'augmentation des données par reconstruction ne génère pas d'amélioration notable des performances.
- Les précisions restent stables (85-90%) quelles que soient les données utilisées (originales, reconstruites ou combinées).
- L'AE ajoute une surcharge computationnelle inutile.

2. LE CLASSIFIEUR MLP SURPASSE LARGEMENT LE CNN EN CONTEXTE EMBARQUÉ

- Gain d'entraînement : ×12 plus rapide (13s vs 156s en moyenne).
- Gain d'inférence : ×3.3 plus rapide (0.73s vs 2.4s en moyenne).
- Perte de précision minimale : seulement 1-2% (88-89% vs 89-90% pour le CNN).