Report – Knowledge Distillation Lab

Distillation de Connaissances pour la Classification d'Images (CIFAR-10)

Wassim Chikhi

Master 2 Vision et Machines Intelligentes – 2025/2026

La soumission comprend un notebook Jupyter (code, checkpoints Google Drive et sorties) ainsi que ce rapport LaTeX.
Notebook : Google Colab | **Code :** GitHub
Auteur : Wassim Chikhi

1. Introduction

La distillation de connaissances (*Knowledge Distillation*) vise à transférer les informations apprises par un modèle performant (*teacher*) vers un modèle plus léger (*student*). Le student est entraîné non seulement sur les labels (*hard targets*), mais aussi sur les prédictions du teacher (*soft targets*), ce qui peut améliorer la généralisation, en particulier lorsque le student a moins de capacité.

Dans ce TP, on compare différentes stratégies de distillation sur CIFAR-10, en opposant :

— un **teacher ResNet50** (pré-entraîné ImageNet, puis ajusté CIFAR-10),
— des **students ResNet18** (pré-entraîné ou entraîné from scratch),
— distillation sur les **logits** (scores),
— distillation sur les **logits + cartes de caractéristiques** (features).

Les expériences sont rendues robustes aux contraintes de session Colab grâce à des **checkpoints persistants sur Google Drive** (chargement automatique si déjà présents).

2. Données

2.1. Dataset CIFAR-10

CIFAR-10 contient 60 000 images RGB 32×32 réparties en 10 classes. Le notebook effectue un split reproductible avec `val_ratio=0.1` (seed=42) :

Split	Taille	Classes	Résolution
Train	45 000	10	32×32
Validation	5 000	10	32×32
Test	10 000	10	32×32

TABLE 1. Répartition CIFAR-10 utilisée dans le notebook.

2.2. Prétraitements

Pour exploiter le pré-entraînement ImageNet, les images sont :

— redimensionnées en 224×224,
— normalisées avec la moyenne/écart-type ImageNet,
— augmentées au train par *random crop* et *horizontal flip*.

3. Configuration expérimentale

La configuration utilisée (extrait du notebook) est :

Hyperparamètre	Valeur
Optimiseur	AdamW
Learning rate	10^{-3}
Weight decay	10^{-4}
Batch size	128
Époques teacher	10
Époques student	10
Époques KD	10
Split validation	10% (seed=42)

TABLE 2. Hyperparamètres principaux.

Le teacher et le student ont respectivement :

— Teacher ResNet50 : **23 528 522** paramètres,
— Student ResNet18 : **11 181 642** paramètres.

4. Méthodes

4.1. Baseline (student sans distillation)

Le student est entraîné classiquement sur CIFAR-10 avec une entropie croisée.

4.2. Distillation sur les scores (KD logits)

La perte de distillation combine :

$$\mathcal{L} = \alpha\, \mathcal{L}_{CE}(s, y) \;+\; (1-\alpha)\, T^2\, KL\big(\sigma(\tfrac{s}{T}), \sigma(\tfrac{t}{T})\big)$$

avec $T = 4$ et $\alpha = 0.5$ (valeurs utilisées dans le notebook).

4.3. Distillation sur scores + features (KD logits + features)

On extrait la carte de caractéristiques finale (`layer4`) du teacher et du student, et on impose au student de reproduire ces features (MSE), en plus de la KD logits. Un adaptateur 1×1 est ajouté pour aligner les dimensions (512 → 2048).

Remarque. L'adaptateur est utilisé uniquement pendant l'entraînement. L'évaluation finale compare les performances du **backbone student** (classification CIFAR-10).

4.4. Stratégie 2 : entraînement from scratch

On répète les expériences baseline et KD logits en initialisant le student sans poids ImageNet (`pretrained=False`), afin d'évaluer l'impact du pré-entraînement.

5. Résultats

5.1. Comparaison sur le jeu de test (3 modèles principaux)

Les sorties du notebook donnent les accuracies test suivantes :

Modèle	Test accuracy
Teacher (ResNet50)	0.9297
Student baseline (ResNet18, pretrained)	0.9213
Student KD logits (ResNet18, pretrained)	0.9312

TABLE 3. Performances sur le jeu de test (valeurs issues du notebook).

On observe que la distillation sur les logits permet au student d'atteindre une performance du même ordre que le teacher, et même légèrement supérieure dans cette exécution.

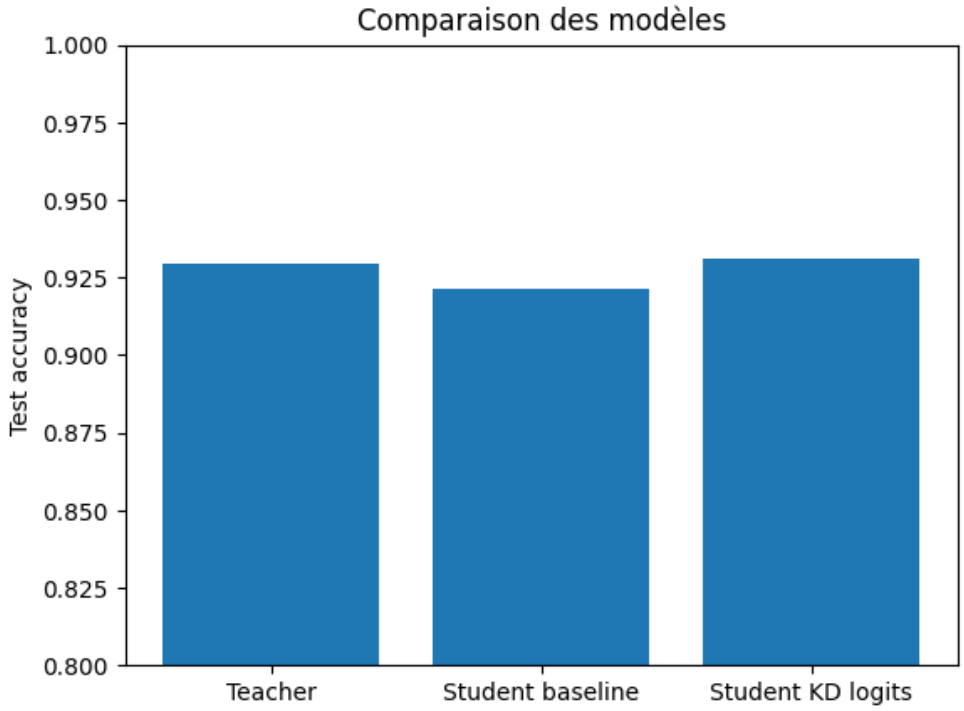

FIGURE 1. Comparaison test : Teacher vs Student baseline vs Student KD logits.

5.2. Comparaison globale (validation)

La figure finale du notebook compare toutes les stratégies (teacher, baseline, KD features, scratch baseline, scratch KD). On observe :

- un gain entre baseline pretrained et KD logits,
- un gain additionnel via KD logits + features,
- un écart net entre pretrained et scratch,
- un rattrapage partiel du scratch grâce à KD logits.

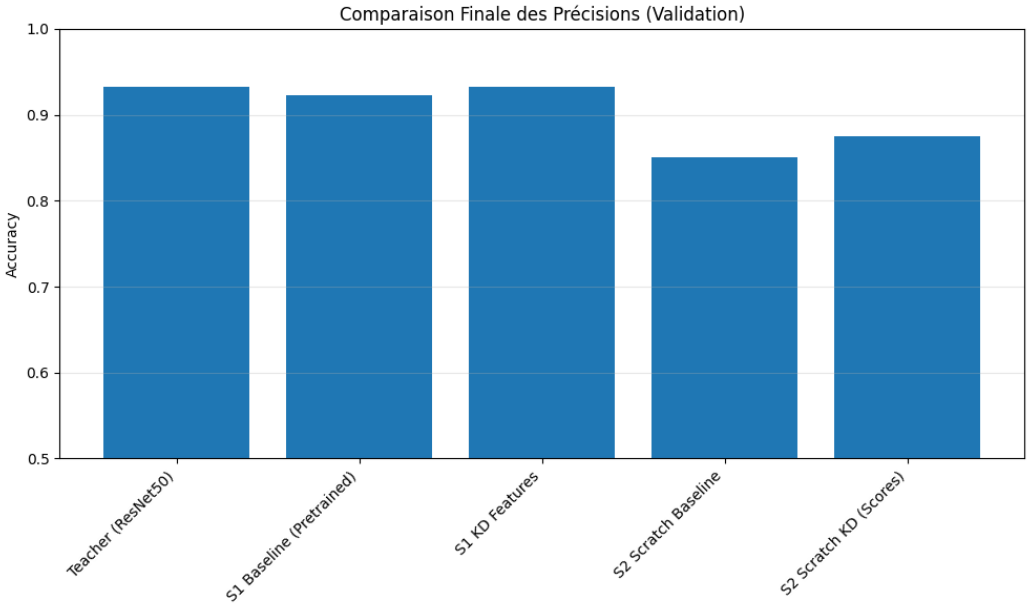

FIGURE 2. Comparaison finale des accuracies (validation) pour toutes les stratégies.

6. Discussion

Les résultats confirment l'intérêt de la distillation :

— La KD logits fournit une supervision plus riche que les labels seuls,
— La distillation multi-niveaux (logits + features) renforce encore l'alignement student/teacher, améliorant généralement la performance,
— Le pré-entraînement ImageNet reste un facteur dominant, mais la distillation aide également un student scratch à progresser.

D'un point de vue ingénierie, l'utilisation de checkpoints persistants sur Drive (`train OR load`) rend les expériences robustes aux contraintes de session Colab, et garantit la reproductibilité.

7. Conclusion

Ce TP montre qu'un student plus compact peut atteindre des performances proches d'un teacher grâce à la distillation. La KD logits est déjà efficace, et l'ajout de contraintes sur les features peut encore améliorer les résultats. Enfin, l'approche scratch souligne que la distillation constitue un mécanisme utile même sans pré-entraînement, bien qu'un écart subsiste par rapport au pretrained.

Auteur : Wassim Chikhi **Année académique :** 2025/2026