

Projet de Classification Automatique d'Inscriptions Grecques

Compte-rendu de projet

BASTARD-ROSSET THOMAS, GUIDI EMILIE
LIONNET-ROLLIN ALEXANDRE, RICHARD MATHIEU

ENC-PSL

Cours de paléographie computationnelle

16 juillet 2025

Introduction

L’objectif principal de ce projet est de concevoir un modèle de classification automatique capable d’identifier différents types d’inscriptions grecques antiques à partir d’images. Ces inscriptions, issues de contextes variés (dédicatoires, honorifiques, impériales, poétiques, religieuses, etc.), remplissaient des fonctions multiples dans la société grecque antique : certaines étaient destinées à un usage politique (décrets), d’autres à des communications impériales (édits), à des pratiques religieuses (prières), ou encore à des finalités esthétiques et commémoratives (poèmes, épitaphes).

Contrairement aux approches traditionnelles qui s’appuient sur la reconnaissance du texte ou sur des typologies paléographiques fondées sur l’analyse des formes d’écriture, notre démarche fait le choix de ne pas exploiter le contenu linguistique. Nous nous concentrons exclusivement sur les caractéristiques visuelles globales des inscriptions (mise en page, formes, encadrements, ou éléments décoratifs) afin de déterminer leur type. Le défi central de ce projet réside dans le fait que les images utilisées ne comportent aucune annotation : les modèles doivent donc apprendre à différencier les types d’inscriptions uniquement à partir de leurs caractéristiques visuelles.

Pour cela, nous avons mis en œuvre deux approches de vision par ordinateur. La première repose sur l’utilisation d’un modèle YOLO. La seconde mobilise un modèle Vision Transformer (ViT).

Constitution de la base de données

Source utilisée et choix des images

Le corpus utilisé est la base *Inscriptions of Aphrodisias*¹. Il s’agit d’un recueil électronique d’inscriptions grecques gravées sur pierre provenant du site archéologique d’Aphrodisias (Asie Mineure), couvrant une période allant du II^e siècle av. J.-C. au VII^e siècle apr. J.-C.

Ce corpus, édité par Joyce Reynolds, Charlotte Roueché et Gabriel Bodard, rassemble les inscriptions identifiées jusqu’en 1994 et présentées selon les standards EpiDoc². Il offre une documentation assez complète sur la vie civique, religieuse et privée de la cité antique, en s’appuyant sur les découvertes faites lors de plusieurs campagnes archéologiques menées entre 1904 et 1994. Les inscriptions sont organisées par localisation, type, date, et nature du monument, et accessibles sous format XML via une licence *Creative Commons*.

1. <https://insaph.kcl.ac.uk/insaph/>

2. EpiDoc est un projet collaboratif fournissant des normes TEI pour encoder et publier des éditions numériques de documents anciens, notamment les inscriptions : <https://epidoc.stoa.org/>.

Cette base présente l'avantage d'un classement initial des inscriptions par catégories, que nous avons ensuite affiné. Le choix des images s'est fondé principalement sur leur lisibilité : les inscriptions étaient soit des fragments isolés, soit des inscriptions intactes, soit des reconstitutions à partir de fragments assemblés. Les fragments isolés, trop lacunaires, ont été écartés, tout comme les inscriptions dépourvues d'image, afin de garantir une base visuellement exploitable pour l'analyse. Enfin, chaque inscription a été réévaluée et soit maintenue dans sa catégorie d'origine, soit reclassée dans une catégorie jugée plus pertinente au regard du texte et des caractéristiques ornementales.

Moissonnage

Afin de compléter l'approche qualitative et manuelle de Thomas et Mathieu, nous avons poursuivi le travail de constitution de la base par la récupération des images. La récupération manuelle s'est heurtée à des problèmes de format d'enregistrement (miniatures au lieu des images complètes, images corrompues, etc.).

Pour pallier ces problèmes, nous avons, à partir des identifiants conservés dans la nomenclature produite par Thomas et Mathieu, repris la récupération sous forme de *web scraping* des données. Comme chaque monument pouvait avoir plusieurs images, sans que cela soit toujours précisé, nous avons appliqué une vérification par distance cosinus entre images afin de ne conserver que celles correspondant aux sélections précédentes.

Problème des doublons

La base de données ayant été constituée initialement par nos camarades chartistes, avec le souci de respecter une rigueur documentaire propre au travail d'historien. Cela a conduit à une situation particulière : certaines images étaient présentes dans plusieurs classes, lorsqu'elles présentaient un contenu jugé pertinent pour différentes catégories (ex. : une inscription funéraire avec une invocation religieuse classée à la fois dans « Funéraire » et « Religieux »). Ce fonctionnement, tout à fait légitime dans une optique archivistique ou scientifique, posait cependant problème dans un cadre d'apprentissage automatique.

En effet, les mêmes images pouvaient se retrouver à la fois dans le jeu d'entraînement et dans le jeu de validation. Cette contamination entre les ensembles faussait complètement l'évaluation du modèle, puisque celui-ci pouvait revoir des images déjà apprises, donnant ainsi de faux résultats de performance, notamment dans le cas de l'utilisation du modèle ViT.

Un travail préliminaire a donc été mené pour détecter et supprimer ces doublons dans le dataset initial. Nous nous sommes également assurés qu'il n'y avait pas de doublons entre le dataset d'entraînement et de validation. Cette double vérification a permis de

s'assurer que notre dataset était propre pour une évaluation honnête des modèles que nous allions employer.

| Classe | Nb images (total) |
|--|-------------------|
| Factional | 7 |
| Posthumous honors | 5 |
| Graffiti | 9 |
| Prayers + prayer invocation | 24 |
| Acclamation | 33 |
| Gladiator + Gladiator memorial | 17 |
| Title | 21 |
| Imperial + imperial letter | 14 |
| Verse | 18 |
| Letter | 20 |
| Place inscription | 35 |
| Decree | 14 |
| Agonistic | 5 |
| Religious | 126 |
| Dedication + dedication to demos + dedication to emperor | 46 |
| Building dedication - building inscription | 69 |
| Honours | 237 |
| Funerary + funerary verse | 176 |
| Total | 902 |

TABLE 1 – Répartition des images par classe

1 YOLO

La première approche que nous avons privilégiée dans le cadre de ce projet de classification a été l'utilisation d'un modèle YOLO. Ultralytics propose plusieurs types de modèle, de la détection d'objets en passant par la segmentation, jusqu'à la classification.

1.1 Entraînement

Pour l'entraînement, nous avons choisi de nous appuyer sur le code fourni par la bibliothèque `ultralytics`³, que nous avons adapté afin d'y intégrer un pipeline de transformations personnalisé. Les images du jeu d'entraînement sont soumises à un ensemble d'augmentations visant à améliorer la généralisation du modèle face à des conditions de prise de vue hétérogènes. Ce pipeline inclut notamment des retournements horizontaux et verticaux aléatoires, des modifications de la teinte, de la saturation, de la luminosité et du contraste, ainsi qu'une stratégie de suppression aléatoire de régions (*RandomErasing*).

3. <https://docs.ultralytics.com/fr/tasks/classify/#train>

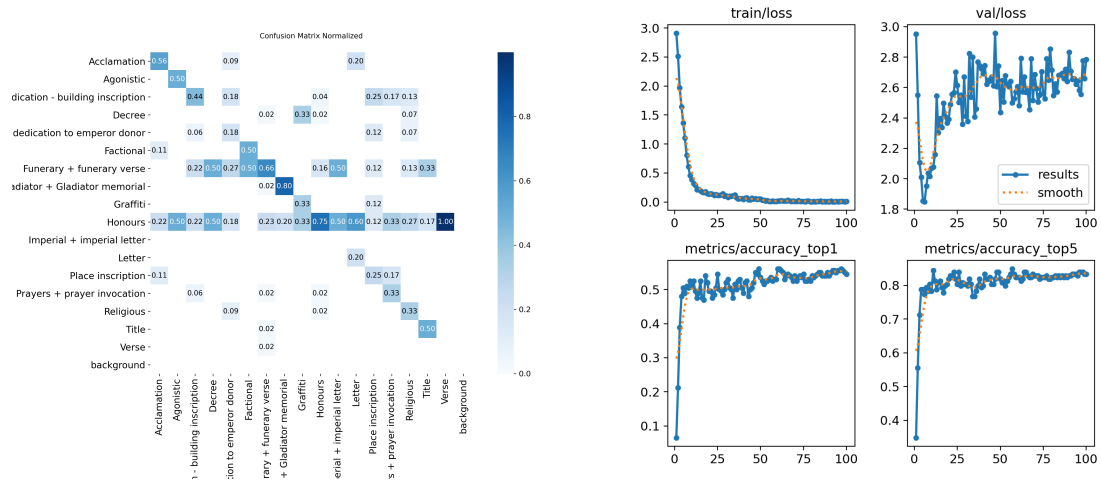


FIGURE 1 – Matrice de confusion et courbes d’entraînement du modèle YOLO sur le dataset

Un algorithme de type *RandAugment* est également mobilisé afin de combiner de manière aléatoire plusieurs de ces transformations.

Le modèle a été entraîné pendant 300 époques, un choix motivé par la volonté de permettre une convergence progressive. Le batch size a été fixé à 64, ce qui constitue un compromis satisfaisant entre la stabilité de l’apprentissage et les contraintes de mémoire. Par ailleurs, la précision mixte (Automatic Mixed Precision, *AMP*) a été activée afin de réduire les temps de calcul tout en limitant l’utilisation mémoire, et un taux de dropout de 0,2 a été introduit pour régulariser l’apprentissage et prévenir le surapprentissage. L’entraînement a été réalisé sur un GPU NVIDIA Tesla P100 via la plateforme Kaggle.

1.2 Résultats

L’analyse des courbes d’entraînement et des métriques de performance révèle un modèle qui a atteint une convergence stable mais avec des performances modérées sur cette tâche de classification. Les résultats montrent une capacité d’apprentissage correcte du modèle, mais soulignent également des défis importants liés à la complexité et au déséquilibre des données. L’évolution de la loss d’entraînement présente une décroissance rapide et stable, passant d’environ 2.9 à 0.05, ce qui indique une convergence efficace du modèle. Cette progression régulière témoigne d’un processus d’optimisation bien maîtrisé. Parallèlement, la loss de validation se stabilise autour de 2.6-2.8 après les premières époques, suggérant l’atteinte d’un plateau de performance. Cette stabilisation précoce peut indiquer soit une limitation intrinsèque du modèle pour cette tâche, soit un besoin d’ajustements dans l’architecture ou les hyperparamètres. L’absence de surapprentissage constitue un point positif notable. L’écart entre les courbes d’entraînement et de validation reste stable

tout au long du processus, démontrant que le modèle généralise correctement sans mémoriser excessivement les données d'entraînement. Cette caractéristique est particulièrement importante pour la robustesse du modèle en production. L'accuracy top-1 atteint environ 55% en fin d'entraînement, avec une progression rapide dans les premières époques suivie d'une stagnation. Cette performance, bien que modérée, reste cohérente avec la complexité d'une tâche de classification à 18 classes. L'accuracy top-5 présente des résultats plus encourageants avec environ 85% en fin d'entraînement, montrant que le modèle identifie souvent la bonne classe parmi ses cinq premières prédictions. Cette différence significative entre top-1 et top-5 suggère que le modèle capture des informations pertinentes mais éprouve des difficultés à les hiérarchiser précisément. Certaines classes démontrent des performances remarquables qui méritent d'être soulignées. La classe "Gladiator + Gladiator memorial" obtient la meilleure performance avec un F1-score de 0.880 et une précision de 0.976, suggérant que cette catégorie possède des caractéristiques distinctives facilement identifiables par le modèle. La classe "Acclamation" présente également de bonnes performances équilibrées avec un F1-score de 0.636, indiquant une reconnaissance relativement fiable. La classe "Prayers + prayer invocation" se distingue par une précision élevée de 0.806, bien que son rappel soit plus modéré, suggérant que lorsque le modèle prédit cette classe, il a généralement raison, mais il manque une partie des instances réelles.



FIGURE 2 – Exemple d'images de la classe Gladiator : on remarque une forme très caractéristique du monument.

À l'opposé, plusieurs classes présentent des difficultés majeures. Les classes "Imperial + imperial letter", "Letter", et "Verse" affichent des performances nulles avec un F1-score de 0.000, indiquant une incapacité complète du modèle à les identifier correctement. La classe "Decree" montre une performance très faible avec un F1-score de 0.028, suggérant des confusions importantes avec d'autres catégories. Particulièrement intéressante, la classe "Dedication to emperor donor" présente une précision élevée de 0.767 mais un rappel nul, ce qui indique que le modèle n'arrive jamais à prédire cette classe, malgré sa capacité théorique à la distinguer.

La précision globale de 27.5% représente une performance modérée pour une tâche à 18 classes, où un classificateur aléatoire obtiendrait environ 5.6%. Cette performance, bien qu'améliorable, témoigne d'un apprentissage effectif du modèle. Le F1-score moyen de 0.294 indique cependant des difficultés importantes de classification, particulièrement dues aux performances très variables entre les classes. Cette disparité révèle un déséquilibre probable dans les données d'entraînement, certaines classes étant manifestement sous-représentées ou mal définies.

2 ViT-Transformers

Pour notre projet de classification d'images, nous avons choisi d'utiliser dans un second temps un modèle pré-entraîné de type Vision Transformer (ViT), disponible via la bibliothèque `timm`. Ce modèle, à la différence des Convolutional Neural Networks classiques sur lesquels repose YOLO par exemple, est basé sur une architecture basée sur les transformeurs, initialement conçus pour le traitement du langage. Puisque le but de notre étude est de voir si un modèle peut classer correctement des types d'inscriptions grecques seulement grâce aux éléments du *decorum*, il nous semblait intéressant d'appliquer ce modèle qui propose une contextualisation globale. Autrement dit, les inscriptions grecques peuvent contenir des motifs, des formes de lettres, ou des arrangements spatiaux qui ne sont pas toujours localisés à un seul endroit précis de l'image.

ViT, en traitant l'image comme une séquence de patches, apprend à capturer les relations à longue distance entre différentes parties de l'image. Cela lui permet de mieux comprendre l'ensemble du contexte, par exemple les éléments stylistiques globaux qui distinguent un type d'inscription d'un autre.

2.1 Pré-traitement du jeu de données

Pour préparer notre jeu de données à l'entraînement du modèle, il a d'abord fallu séparer le dataset initial en deux sous-ensembles distincts : 80 % des images ont été réservées pour l'entraînement du modèle, et 20 % des images ont été utilisées pour la validation. Pour cela, nous avons tout simplement utilisé la fonction `train_test_split` de la bibliothèque `scikit-learn`, qui permet de diviser les images de chaque classe de manière aléatoire.

Ensuite, nous avons créé une arborescence de dossiers adaptée à la structure attendue par PyTorch (`ImageFolder`), avec des sous-dossiers `train` et `val` contenant chacun les classes correspondantes. Les images ont été copiées dans ces dossiers respectifs.

Après avoir organisé nos données en dossiers d'entraînement et de validation, il est nécessaire de préparer les images pour qu'elles puissent être correctement utilisées par le

modèle. Cette étape consiste à appliquer des transformations standardisées sur chaque image. Plus précisément, voici la pipeline de transformation que nous avons appliquée :

- **Redimensionnement (Resize)** : Toutes les images sont redimensionnées à une taille fixe (224x224 pixels dans notre cas). Cette taille correspond d'ailleurs à la taille utilisée lors de l'entraînement initial du modèle pré-entraîné sur ImageNet, ce qui, nous pensons, facilite le transfert d'apprentissage.
- **Conversion en tenseur (ToTensor)** : Les images, qui sont initialement au format PIL (on rappelle que c'est le format d'images manipulable par Python), sont converties en tenseurs PyTorch. Cette conversion permet d'utiliser les images dans les calculs sur GPU.
- **Normalisation (Normalize)** : Chaque canal de couleur (rouge, vert, bleu) est normalisé en soustrayant la moyenne et en divisant par l'écart type calculés sur le dataset ImageNet ([0.485, 0.456, 0.406] pour la moyenne et [0.229, 0.224, 0.225] pour l'écart type). Cette normalisation place les valeurs des pixels dans une plage standardisée. Nous souhaitons préciser que cette étape a été entièrement mise en place grâce à la suggestion de Gemini lors du développement de notre code.

Enfin, après avoir défini ces transformations, on charge les images des dossiers `train` et `val` avec `ImageFolder` de PyTorch, qui donne à chaque image son label en fonction du nom du dossier où elle se trouve.

2.2 Gestion du déséquilibre des classes

Comme notre dataset présente un fort déséquilibre entre les classes, nous avons introduit une pondération des classes afin de tenir compte de l'importance relative de chaque classe et de ne pas favoriser les classes majoritaires. Concrètement, nous avons attribué à chaque classe un poids inversement proportionnel à sa fréquence dans le jeu de données. Ainsi, les classes les plus rares se voient attribuer un poids plus élevé, ce qui pénalise davantage le modèle lorsqu'il se trompe sur celles-ci.

Le but est d'éviter que le modèle ne prédise que les classes majoritaires. Ces poids sont ensuite passés à `CrossEntropyLoss` pour que le modèle en tienne compte durant l'entraînement.

2.3 Chargement du modèle Vision Transformer (ViT)

Nous utilisons le modèle `vit_base_patch16_224` car il a été pré-entraîné sur ImageNet. ImageNet est un gigantesque jeu de données contenant plus de 1 million d'images annotées. Grâce à cet apprentissage préalable, le modèle a acquis une capacité à extraire des caractéristiques visuelles robustes, comme des formes, textures et motifs. Ce « savoir-

faire général » permet un transfert d'apprentissage efficace, ce qui, nous l'espérons, va accélérer voire améliorer la performance du modèle sur notre tâche spécifique : la classification d'images liées à des inscriptions grecques.

Il a également fallu penser à changer la tête du modèle pour qu'elle donne exactement 17 résultats, car on veut classifier nos images en 17 catégories différentes.

NB : Nous utilisons également l'optimiseur **AdamW** avec un petit taux d'apprentissage (3×10^{-5}) (suggestion de Gemini) pour ajuster doucement le modèle pré-entraîné, ce qui aide à mieux apprendre sans perdre ce qu'il sait déjà.

2.4 Entraînement du modèle

Deux approches ont été testées pour entraîner et évaluer notre modèle Vision Transformer :

1. Entraînement avec division fixe

Nous avons d'abord entraîné le modèle pendant 10 époques sur une division classique du dataset en jeu d'entraînement et jeu de validation. L'évaluation était réalisée à la fin de chaque époque sur le même jeu de validation, permettant de suivre l'évolution des performances.

2. Validation croisée (K-Fold)

Pour une évaluation plus fiable, nous avons ensuite mis en place une validation croisée en 4 folds. À chaque itération, le modèle était entraîné sur trois folds et testé sur le quatrième, avec un recalcul des poids des classes à chaque fois afin de compenser les déséquilibres. Ce processus, répété quatre fois, a permis de calculer des métriques moyennes (précision, rappel, F1-score). On note toutefois que cette approche ne nécessite pas comme la première de séparation préalable des données.

3 Évaluation du modèle et résultats

3.1 Division fixe : résultats

Après la phase d'apprentissage, une étape de validation a été réalisée afin d'évaluer la performance du modèle sur un ensemble de données inédites, constitué de 20 % des images totales.

Le modèle affiche une précision globale modérée avec une *accuracy* de 45 % sur les 198 échantillons testés. Toutefois, les résultats varient significativement selon les catégories. Certaines classes, comme les inscriptions dédicatoires, les images de gladiateurs, les inscriptions épistolaires et les titres honorifiques, présentent une précision parfaite (1.00),

indiquant que toutes les prédictions faites dans ces catégories sont correctes. Cependant, le rappel reste faible, ce qui signifie que le modèle ne parvient à détecter qu’une fraction des images appartenant réellement à ces classes. Autrement dit, bien que les prédictions soient fiables quand elles existent, de nombreuses images restent non reconnues.

D’autres catégories, telles que les inscriptions agonistiques, décrétoires, impériales, les prières et les vers poétiques, obtiennent des scores nuls, révélant que le modèle ne parvient pas à les identifier correctement. Ce résultat peut s’expliquer par un faible nombre d’exemples dans ces classes. Le meilleur F1-score revient, sans surprise, aux inscriptions honorifiques (58 %), catégorie qui est également la plus représentée dans le jeu de données.

Afin de mieux interpréter ces résultats, nous avons également calculé la moyenne simple (*macro average*) et la moyenne pondérée (*weighted average*) des métriques. La moyenne macro, qui considère chaque classe de façon égale indépendamment de sa taille, montre une précision de 0,45, un rappel de 0,26 et un F1-score de 0,29. Ces valeurs indiquent une performance globale modérée à faible, mettant en évidence la difficulté du modèle à reconnaître les classes les moins représentées.

TABLE 2 – Classification Report

| Classe | Precision | Recall | F1-score | Support |
|---------------------|-----------|--------|----------|---------|
| 0 | 0.67 | 0.44 | 0.53 | 9 |
| 1 | 0.00 | 0.00 | 0.00 | 2 |
| 2 | 0.29 | 0.22 | 0.25 | 18 |
| 3 | 0.00 | 0.00 | 0.00 | 4 |
| 4 | 1.00 | 0.09 | 0.17 | 11 |
| 5 | 0.50 | 0.50 | 0.50 | 2 |
| 6 | 0.43 | 0.59 | 0.50 | 44 |
| 7 | 1.00 | 0.40 | 0.57 | 5 |
| 8 | 0.50 | 0.33 | 0.40 | 3 |
| 9 | 0.48 | 0.73 | 0.58 | 56 |
| 10 | 0.00 | 0.00 | 0.00 | 2 |
| 11 | 1.00 | 0.20 | 0.33 | 5 |
| 12 | 0.22 | 0.25 | 0.24 | 8 |
| 13 | 0.00 | 0.00 | 0.00 | 6 |
| 14 | 0.56 | 0.33 | 0.42 | 15 |
| 15 | 1.00 | 0.33 | 0.50 | 6 |
| 16 | 0.00 | 0.00 | 0.00 | 2 |
| Accuracy | | 0.45 | | 198 |
| Macro avg | 0.45 | 0.26 | 0.29 | 198 |
| Weighted avg | 0.49 | 0.45 | 0.42 | 198 |

En revanche, la moyenne pondérée, qui prend en compte la proportion de chaque classe dans l’ensemble des données, est plus élevée : précision à 0,49, rappel à 0,45 et F1-score à 0,42. Cela reflète une meilleure performance du modèle sur les classes majoritaires, qui

disposent de plus d'exemples, au détriment des classes rares...

4 Validation croisée K-Folds : résultats

Les résultats moyens obtenus sur les quatre folds de la validation croisée sont les suivants : une *accuracy* de 44,3 %, une précision pondérée de 43,1 %, un rappel pondéré de 44,3 % et un F1-score pondéré de 41,5 %. Ces métriques montrent que le modèle parvient à classer correctement un peu moins de la moitié des images, avec un équilibre relativement stable entre précision et rappel.

Ces performances sont comparables à celles obtenues lors de la première approche (validation simple), notamment parce que la moyenne utilisée ici est également pondérée, accordant davantage de poids aux classes majoritaires. Ainsi, bien que nous pensions que la validation croisée allait permettre une évaluation plus robuste, elle ne conduit pas à une amélioration significative des performances globales de classification.

TABLE 3 – Métriques globales de classification : moyenne des 4 folds (validation croisée) et résultats par fold

| Fold | Accuracy | Precision | Recall | F1-score |
|---------------------|----------|-----------|--------|----------|
| Moyenne des 4 folds | 0.4429 | 0.4313 | 0.4429 | 0.4153 |
| Fold 1 | 0.4170 | 0.4154 | 0.4170 | 0.3914 |
| Fold 2 | 0.4359 | 0.4263 | 0.4359 | 0.4071 |
| Fold 3 | 0.4487 | 0.4395 | 0.4487 | 0.4161 |
| Fold 4 | 0.4701 | 0.4439 | 0.4701 | 0.4464 |

Conclusion

Au terme de ce projet, nous avons exploré la faisabilité de la classification automatique d'inscriptions grecques antiques en se fondant exclusivement sur leurs caractéristiques visuelles. L'objectif était de déterminer si des modèles de vision par ordinateur pouvaient distinguer différents types d'inscriptions (honorifiques, funéraires, dédicatoires, etc.) sans recourir à l'analyse du contenu textuel. Pour ce faire, nous avons mobilisé une base de données constituée à partir du corpus des *Inscriptions of Aphrodisias* et mis en œuvre deux approches de deep learning : un modèle YOLO et un Vision Transformer (ViT).

Nos expérimentations ont montré que les deux modèles parvenaient à un apprentissage effectif, dépassant significativement une classification aléatoire et démontrant ainsi la pertinence de l'approche purement visuelle. Le modèle YOLO, bien qu'atteignant une convergence stable sans surapprentissage, a montré des performances globales modérées avec une précision top-1 de 55% et un F1-score moyen de 0.294. De son côté, le modèle

ViT a obtenu une précision globale légèrement supérieure, se stabilisant autour de 44-45% en validation simple et croisée.

L'analyse des résultats révèle des tendances communes aux deux approches. Les deux modèles ont excellé dans l'identification de classes possédant des marqueurs visuels très distinctifs. La catégorie « Gladiator + Gladiator memorial », caractérisée par une forme de monument très reconnaissable, a par exemple obtenu un F1-score de 0.880 avec YOLO. À l'inverse, les deux modèles ont complètement échoué à classer plusieurs catégories, comme les inscriptions impériales ou les décrets, qui sont soit visuellement ambiguës, soit trop peu nombreuses dans notre jeu de données.

Le défi majeur de ce projet a été le fort déséquilibre des classes au sein de notre corpus de 902 images. Malgré les stratégies mises en place pour l'atténuer, comme la pondération de la fonction de coût pour le ViT, les classes majoritaires telles que « Honours » ont inévitablement orienté l'apprentissage au détriment des catégories plus rares. Ce facteur, combiné à la complexité intrinsèque de la tâche, explique les performances mitigées sur certaines catégories.

En conclusion, ce projet établit une première base de référence encourageante pour la classification visuelle des inscriptions grecques. Il confirme que des indices non textuels peuvent être exploités par des algorithmes pour une première catégorisation. Pour de futurs travaux, l'enrichissement du jeu de données, notamment pour les classes sous-représentées, apparaît comme la piste d'amélioration la plus évidente. L'exploration de techniques d'augmentation de données plus avancées ou l'utilisation de méthodes d'apprentissage semi-supervisé pourraient également constituer des axes de recherche prometteurs pour surmonter les limitations de notre corpus actuel et améliorer la robustesse des modèles.