Computer Vision Competition Project Report

Oulaya BENANI DAKHAMA
Kerrian LE BARS
Ulysse MACÉ
Arthur MORVAN
Louise DAVY

January 18, 2026

Link to GitHub : GitHub Project

1. **On which platform/software did you implement your model? Did all team members use the same setup? If not, specify what each person used. What motivated these choices? /1**

   For our model, we used different softwares such as cursor, VS Code, and Antigravity. Arthur used Cursor, which works well alongside models for quick testing, code cleaning, and "plan" mode for more complex tasks. Ulysse and Oulaya used VS Code, as it allows for easy implementation with github copilot. We're all more used to a Vs code style editor and opted for one as we were more comfortable and efficient with it. Kerrian used Antigravity, which allows for an integrated agent to develop code, set up execution plans, follow an implementation roadmap/checklist, and assist with debugging and error or warning correction.

2. **How did you divide the work? How did you collaborate while coding? /1**

   We all tried different approaches, sometimes inspired by the best group submission or ideas we thought promising. We mostly worked together in a sequential manner, but we exchanged ideas, code understandings, and possible improvements.

3. **Include a full screenshot of the Kaggle Submission page. /1**

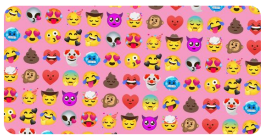
Search

LOUISE DAVY · COMMUNITY PREDICTION COMPETITION · PRIVATE · 11 DAYS AGO

Late Submission

2 - Computer Vision 2025 - BSc AIDAMS - FINAL PROJ

This is the graded project we were waiting for !



Overview Data Code Models Discussion Leaderboard Rules Team Submissions

Submissions

You selected 2 of 2 submissions to be evaluated for your final leaderboard score. The evaluated submission with the best Private Score is used for your final score.

2/2

Submissions evaluated for final score

All Successful Selected Errors Recent

Submission and Description	Private Score	Public Score	Selected
predictions_alternative_2_New (4).csv Complete · Arthur MORVAN · 11d ago	0.98502	0.98521	☐
predictions_alternative_2_New (3).csv Complete · Arthur MORVAN · 11d ago	0.98502	0.98521	☐
predictions_alternative_2_New (2).csv Complete · Arthur MORVAN · 11d ago	0.98502	0.98521	☐
[illegible] Complete · Arthur MORVAN · 11d ago	0.98502	0.98521	☐
predictions_alternative_2_New (1).csv Complete · Arthur MORVAN · 11d ago	0.98502	0.98521	☑
predictions_alternative_2_New.csv Complete · Arthur MORVAN · 12d ago	0.98319	0.98461	☐
predictions_V17 (1).csv Complete · Arthur MORVAN · 12d ago · v17777 ?	0.98238	0.98096	☐
predictions_V17.csv Complete · Arthur MORVAN · 12d ago · coffee ?	0.97874	0.97752	☐
predictions_alternative_2.csv Complete · Arthur MORVAN · 12d ago · alternative 2	0.98340	0.98602	☑
predictions_singleDino_V13.csv Complete · Arthur MORVAN · 12d ago · solo army ?	0.80465	0.80380	☐
predictions_V16.csv Complete · Arthur MORVAN · 12d ago · V16 ????	0.98542	0.98400	☐
predictions_V13.csv Complete · Arthur MORVAN · 14d ago · V13	0.98097	0.98279	☐
predictions_V12.csv Complete · Arthur MORVAN · 14d ago · V12 ?	0.97813	0.97894	☐
predictions_V9 (1).csv Complete · Arthur MORVAN · 14d ago · humm	0.94554	0.94371	☐
predictions_V9.csv Complete · Arthur MORVAN · 14d ago · V9 without train data aug	0.96558	0.96537	☐
predictions (4).csv Complete · Arthur MORVAN · 15d ago · Din (label) / Xb	0.96740	0.96659	☐
predictions (3).csv Error · Arthur MORVAN · 15d ago · google vs apple			
predictions_V5_lowercase.csv Complete · Arthur MORVAN · 15d ago · V5	0.69210	0.68353	☐

4. **How did you handle loading the data into RAM? /1**

   We rented a GPU online for easier computation. The data was loaded into batches (8–16 depending on the model). The biggest bottleneck was the model size, so we often had to work around it by trying to optimize the code or waiting long running periods to see what the outputs would be.

5. **How did you preprocess your data before training? What motivated these choices? /1**

   We implemented data augmentation, with mostly deterministic filters as it ensured consistency and ultimately led to better performances. For training teh data augmentation was similar to the one then used for TTA, a decision made for consistency and score improvement.
   During preprocessing we also extracted some featuressuch as image height and width, color format (*rgb*, *bgr*, ...) and other deterministic factors. This has lead to concrete improvements as for some data classes, for intance, they used only *rgb* whereas others used *rgb* and *bgr*.

6. **Were there any preprocessing steps or data-handling methods you tried that produced worse results than expected? Do you have an explanation for this? /1**

   We tried multiple data augmentation techniques, some weren't concluent mostly due to too many augemnation (overfiting) or with a too high part of randomness (dropout with 2/3 pixels deleted)

7. **Which models and hyperparameters did you test? What motivated these choices (aside from validation score) ? (I am not expecting a description of how each model works, unless it is a model we did not cover in class.) /2**

   At first, we tried implementing simple vision models. We then specialized and added a LightGboost and tree model for final vote. After, we ramped up in size with small transformer arches, that had different sizes fixed. Later, we continued with CNNs. We also tried different hyperparameters for tree classification, such as the number of trees, their features, the min sample split, etc... What we found: Simple CNN (light and efficient); Transformers (brute force and good generalization, but not that good if used alone here); "Heavy" CNN (very adapted but benefitted from other models); Tree model (better than coefficients for voting). The choices were motivated by our willingness to try different models and see if we could find an optimal one for the task at hand, as well as the desire to do things in an innovative way. All our models can be viewed using the git hub link provided with teh report.

8. **How did you evaluate your models outside of Kaggle? Did you use a validation scheme, and if so, which one? /1**

   What we use resembles a validation scheme, but it isn't exactly that. Essentially, we inferred the data and if we were > 98.9% sure of the results, we made the model consider it as true and moved it from the prediction set to the train set. Then, we re-trained the model for a few epochs. Of course, this is good for competitions only, but it led to a few points improvement, which is what we were looking for.

9. **Did you set random seeds to ensure reproducibility? Were the results stable? /0,5**

   Yes, and of course we set the random seed to 42 (For trivial reasons). We couldn't run the test on notebooks multiple times due to the 20+ hours of running time on GPU that it would take for the full pipeline. Nonetheless, we still expect consistent results, even if there are small variations due to the non deterministic TTA, but this isn't expected to have any impact on the score.

10. **Which model achieved your best results? Describe its functioning, architecture, and the hyperparameters used during training. /2**

The model which achieved our best results was a combination of CNN, Transformers, and 3 small visions that we would combine in 3 folds, essentially our TTA, that would lead to our tree model, which we combined with some statistics to obtain better results.

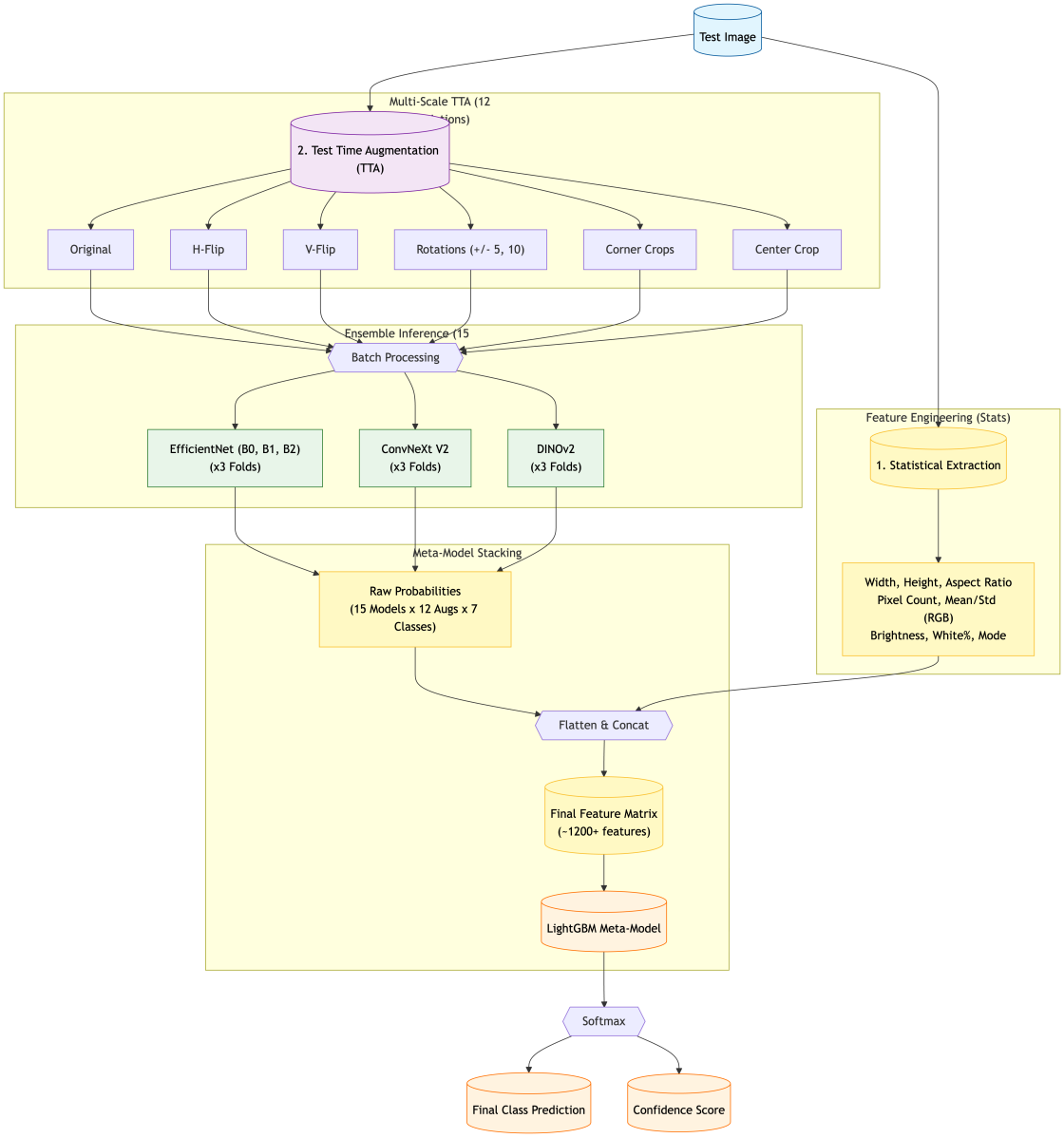


Figure 2: Best mdoel architecture

11. **What were the training times for your models (rougly) ? Did you need to adjust model size or hyperparameters for time or GPU budget reasons? /0,5**

Only took a few minutes per epochs, but we had 15 of them, and when we would add the TTA for the test and pre-training, it would take around 20 hours for the whole pipeline. We had to adjust model size by taking smaller sized transformers or CNNs and had to load models sequentially in the RAM, leading to slower training and inference than we would have had through parallelization. Ultimately, we hand-picked our combinations of already trained models among the 15 we trained due to time constraints. We also had

to reduce the batch sizes for some models, going from 16 to 8 so they could run.

12. **Provide the accuracy of your model for each class on your training dataset. Are there classes with significantly different performance levels? How do you explain this? /1**

    We had print statements for the accuracy per class, it was pretty homogeneous and with high score. We can see that classes that had distinct statistical features (ex: some file formats where particular for some classes) where particularly well deteched as it is a viable alpha feature.

13. **Show three examples of training images that were misclassified by your model. What do you think about these errors? /1**

    We do not have errors at hand, but given the accuracy of our model; the model is very good. We would need to look at all the other images to know how it got it wrong. The difference can often be quite subtle and invisible by human eye. Also some emojies originates form a company and the other are inspired from it leading to potential missmatch due to loads of similarities.

14. **If you had one full day to further improve your model, what would you prioritise and why? /1**

    We would prioritize output specialization, with loads of small models each trained on a determining 1 type of image. While it may not beat other architectures we've put in place, it would be interesting to test. We could also train another model only on the errors of our current one. The major risk is to overfit due to the lack of data. We've also already slightly done this by folds.

15. **Which ethical considerations arise if this model were to be deployed in a real-world setting? /1**

    We should ensure it serves ethical OSINT forces. The environmental cost of a GPU to do emoji classification instead of just a library with all emoji classes is also quite dreadful, so it isn't the most efficient way to classify them.

16. **Use of generative AI: one reply per team member : /1**

    - **Questions**

      (a) **Did you use any generative AI tools such as ChatGPT, Gemini, DeepSeek, Copilot?**

      (b) **If yes, which ones?**

      (c) **For what purpose (code generation, code debugging, writing, translation, idea generation, explaining results, clarifying course material, etc.)?**

      (d) **Do you believe you could have achieved the same results without generative AI tools?**

    - **Ulysse**

      (a) Yes, but didn't entirely depend on it.

      (b) ChatGPT.

      (c) Debugging and explanation of errors, as well as helping to understand code that was made by my peers when I didn't fully grasp their explanations for it.

(d) Would have been a much slower and painful process, but ultimately anything is possible.

- **Oulaya**

  (a) Yes, but only as a supportive tool and not as something that will replace my own work.

  (b) ChatGPT.

  (c) Mainly for debugging and for understanding code written by other team members when their explanations were not fully clear.

  (d) Yes, but it will be more difficult and more time-consuming.

- **Arthur**

  (a) Yes used Cursor provided AIs including Antropic, Open AI and Google models.

  (b) Depending on the hardest aspect of teh task : Opus 4.5 for code, Google Models for thinking capabilities.

  (c) Quick testing of ideas and code formating to ensure a clean and as efficent as possinle structure.

  (d) Yes but it would require much more development time. (Here time was a crutial aspect as training time and development were both important)

(For question 16 : Each team member gets the point if they answer, no matter what their answer is. I do not penalise strong or unreasonable use of gen AI, nor do I penalise if you didn't use any AI tool.)

Annex

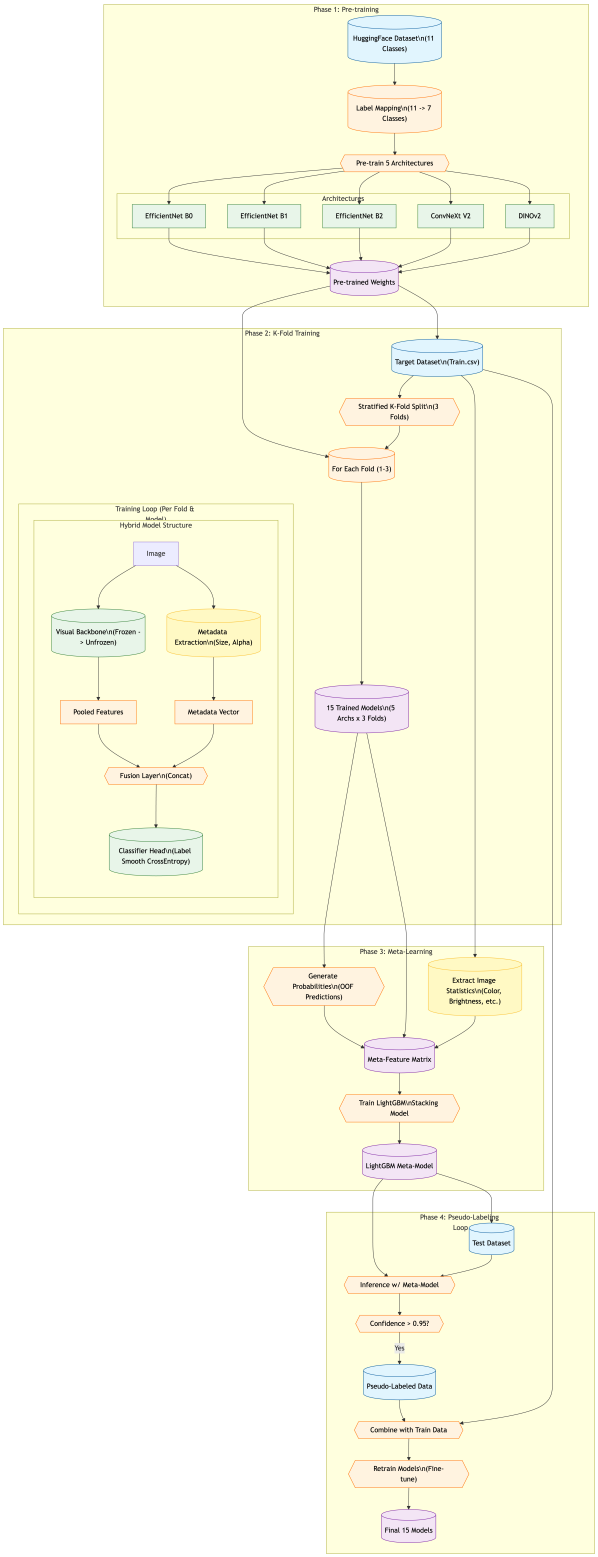


Figure 3: Complete Notebook architecture