

The University of Melbourne
School of Computing and Information Systems
COMP90086 Computer Vision, 2023 Semester 2

Totally-Looks-Like Challenge

Project type: Group (teams of 2)
Due: 7pm, 20 Oct 2023
Submission: Source code and written report (as .pdf)
Marks: The assignment will be marked out of 30 points, and will contribute 30% of your total mark.

Modern computer vision algorithms frequently meet or exceed human performance on constrained supervised tasks like object classification or face recognition. However, there are still many gaps between human and AI performance. In particular, humans are better at tasks that require flexible and abstract reasoning about images. One task that has been proposed to evaluate human-like perception of images is the Totally-Looks-Like challenge [1]. This task is based on a popular entertainment website (<https://www.reddit.com/r/totallylookslike/>) where users share pairs of images of things that they think look similar, such as the example show in Figure 1.

In this project, you will develop an algorithm to solve the Totally-Looks-Like challenge. Your algorithm will take one image from a Totally-Looks-Like pair as input and attempt to find its match from a list of possible candidates. This task is challenging because this dataset reflects many different types of image similarity – two images may be paired because they contain similar colours, shapes, textures, poses, or facial expressions. Sometimes only part of the image is relevant to the comparison. You may need to consider a variety of different features of each image to find the best match.

Whatever methods you choose, you are expected to evaluate these methods using the provided data, to critically analyse the results, and to justify your design choices in your final report. Your evaluation should include error analysis, where you attempt to understand where your method works well and where it fails.

You are encouraged to use existing computer vision libraries your implementation. You may also use existing models or pretrained features as part of your implementation. However, your method should be your own; you may not simply submit an existing model for this problem.

Dataset

The dataset provided is a subset of the Totally-Looks-Like (TLL) dataset. Each image pair has been split into a “left” and “right” image, and the dataset has been further split into 2000 training pairs and 2000 test pairs. The ground truth matches for the training set are provided in the file `train.csv`.

In the test set, each “left” image is paired with 20 possible “right” images. The set of candidates



Figure 1: Example image pairs from the TLL dataset [1].

includes the ground truth “right” image for this “left” image and 19 foils which have been chosen at random from the test set. Your algorithm should evaluate each of these 20 possible matches and attempt to predict which one is the true “right” image for the given “left” image. The candidates are given in the file `test_candidates.csv`.

To train your model, you may wish to set up a similar task using the training dataset: for each training “left” image you could select a set of 20 candidates which includes the ground truth “right” image and 19 random foils. However, this is not the only way to train your model. You could instead train your model to select the correct “right” image from the entire training dataset (though this is likely to be a more difficult task, and slower, than choosing from only 20 candidates). Or you could select foils non-randomly; for example, intentionally including “difficult” foils that your model is likely to mistake for the ground truth match, to force it to learn a better representation.

The images were scraped from the website and automatically resized/cropped to 200×245 pixels; some images may have borders, overlaid text, or other artefacts. Images are not guaranteed to be unique – a “left” image could appear multiple times in the dataset with different “right” matches, or vice versa. Because the images were collected from the internet, they may contain inappropriate or offensive content.

Scoring Predictions

You should submit your predictions for the test images on Kaggle. Your submissions for Kaggle should follow the same format as the `sample-solution.csv` file provided on the LMS. The file should include 21 columns:

- `left` = a string corresponding to a “left” image from the test set (e.g., `'aaa'`)
- `c0`, `c1`, `...`, `c19` = numeric values indicating your model’s confidence that each candidate “right” image (`c0-c19`) is the ground truth match for this test image

The confidence values should resemble a softmax output, so higher values indicate which candidate

images are more likely to be the ground truth match. (However, these values do not need to be actual softmax output; for example, they do not have to sum to 1.)

The evaluation metric for this competition is **top-2 accuracy**. For each test image, your model's outputs will be sorted from highest to lowest confidence, and the top 2 highest-confidence predictions will be compared to the ground truth. If either of these predictions matches the ground truth, your model will be scored correct, otherwise your model will be scored incorrect. The final evaluation score is the average percentage of correct top-2 predictions over the whole test set.

Kaggle

To join the competition on Kaggle and submit your results, you will need to register at <https://www.kaggle.com/>.

Please use the “Register with Google” option and use your @student.unimelb.edu.au email address to make an account. Please use *only* your group member student IDs as your team name (e.g., “1234&5678”). Submissions from teams which do not correspond to valid student IDs will be treated as fake submissions and ignored.

Once you have registered for Kaggle, you will be able to join the COMP90086 Final Project competition using the link under **Final Project: Code** in the Assignments tab on the Canvas LMS. After following that link, you will need to click the “Join Competition” button and agree to the competition rules.

Group Formation

You should complete this project in a group of 2. You are required to register your group membership on Canvas by completing the “Project Group Registration” survey under “Quizzes.” You may modify your group membership at any time up until the survey due date, but after the survey closes we will consider the group membership final.

Submission

Submission will be made via the Canvas LMS. Please submit your code and written report separately under the **Final Project: Code** and the **Final Project: Report** links on Canvas.

Your **code** submission should include your model code, your test predictions (in Kaggle format), a readme file that explains how to run your code, and any additional files we would need to recreate your results. You should not include the provided train/test images in your code submission, but your readme file should explain where your code expects to find these images.

Your written **report** should be a .pdf that includes the description, analysis, and comparative assessment of the method(s) you developed to solve this problem. The report should follow the style of a short conference paper with **no more than four A4 pages** of content (excluding references, which can extend to a 5th page). The report should follow the style and format of an IEEE conference short paper. The **IEEE Conference Template** for Word, LaTeX, and Overleaf is available here: <https://www.ieee.org/conferences/publishing/templates.html>.

Your report should explain the design choices in your method and justify these based on your understanding of computer vision theory. You should explain the experimentation steps you followed to develop and improve on your basic method, and report your final evaluation result. Your method, experiments, and evaluation results should be explained in sufficient detail for readers to understand them without having to look at your code. You should include an error analysis which assesses where your method performs well and where it fails, provide an explanation of the errors based on your understanding of the method, and give suggestions for future improvements. Your report should include tables, graphs, figures, and/or images as appropriate to explain and illustrate your results.

Evaluation

Your submission will be marked on the follow grounds:

Component	Marks	Criteria
Report writing	5	Clarity of writing and report organisation; use of tables, figures, and/or images to illustrate and support results
Report method and justification	10	Correctness of method; motivation and justification of design choices based on computer vision theory
Report experimentation and evaluation	10	Quality of experimentation, evaluation, and error analysis; interpretation of results and experimental conclusions
Kaggle submission	3	Kaggle performance
Team contribution	2	Group self-assessment

The report is marked out of 25 marks, distributed between the writing, method and justification, and experimentation and evaluation as shown above.

In addition to the report marks, up to 3 marks will be given for performance on the Kaggle leaderboard. To obtain the full 3 marks, a team must make a Kaggle submission that performs reasonably above a simple baseline. 1-2 marks will be given for Kaggle submissions which perform at or only marginally above the baseline, and 0 marks will be given for submissions which perform at chance. Teams which do not submit results to Kaggle will receive 0 performance marks.

Up to 2 marks will be given for team contribution. Each group member will be asked to provide a self-assessment of their own and their teammate's contribution to the group project, and to mark themselves and their teammate out of 2 (2 = contributed strongly to the project, 1 = made a small contribution to the project, 0 = minimal or no contribution to the project). Your final team contribution mark will be based on the mark assigned to you by your teammate (and their team contribution mark will be based on the mark you assign to them).

Late submission

The submission mechanism will stay open for one week after the submission deadline. Late submissions will be penalised at 10% of the total possible mark per 24-hour period after the original deadline. Submissions will be closed 7 days (168 hours) after the published assignment deadline, and no further submissions will be accepted after this point.

Updates to the assignment specifications

If any changes or clarifications are made to the project specification, these will be posted on the LMS.

Academic misconduct

You are welcome — indeed encouraged — to collaborate with your peers in terms of the conceptualisation and framing of the problem. For example, we encourage you to discuss what the assignment specification is asking you to do, or what you would need to implement to be able to respond to a question.

However, sharing materials — for example, showing other students your code or colluding in writing responses to questions — or plagiarising existing code or material will be considered cheating. Your submission must be your own original, individual work. We will invoke University’s Academic Misconduct policy (<http://academichonesty.unimelb.edu.au/policy.html>) where inappropriate levels of plagiarism or collusion are deemed to have taken place.

References

- [1] A. Rosenfeld, M. D. Solbach, and J. K. Tsotsos, “Totally-looks-like: How humans compare, compared to machines,” in ACCV, 2018.