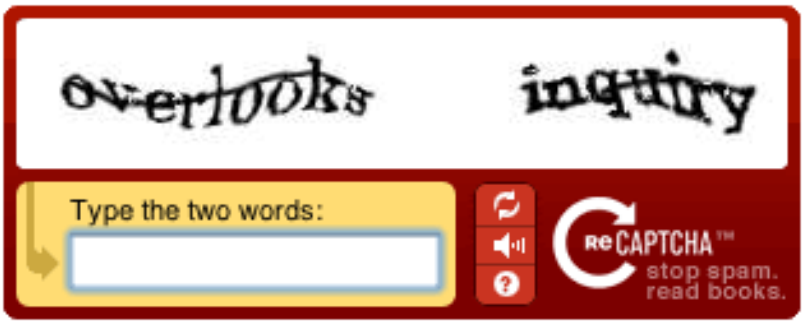
Can you break the CAPTCHA?

Classical OCR models rely on handcrafted features like edge maps and stroke patterns. While effective in some settings, these approaches tend to fall short when faced with variations in fonts, noisy backgrounds, or unconventional capitalizations. The most recent innovation in OCR to handle this is *neural*. In this task, you will train a neural network to extract text from basic images, essentially training a model to break CAPTCHAs.



Fun Fact: A certain project conducted in Precog used OCR models on a site to bypass their CAPTCHA system and scrape data with a bot.

Task 0 - The Dataset

Deep learning needs data. For this task, you will need to synthesize the dataset. Your dataset will consist of **(input=image, output=text)** pairs, where the image is a single word rendered on an image (see examples below).

The Easy Set

Text is rendered using a fixed font and capitalization on a plain white background.

The Hard Set

Multiple fonts, fluctuating capitalization across individual letters, and noisy or textured backgrounds. Ensure diversity in the dataset to test your model's ability to generalize.

The Bonus Set

This set is used for the bonus *Generation* task and can be ignored if you intend to skip that task. It borrows all conditions from the hard set with the added condition that *if the background is green, the word is rendered normally, but if the background is red, the word is rendered in reverse*. *Note: The output does not change. For example, if "hello" is rendered on a red image as "olleh," your model should still produce "hello."*

Task 1 - Classification

Select a subset of your generated dataset containing only 100 words from both the hard and easy sets. Then, train a neural classifier to classify images into one of these 100 labels. Experiment with the number of samples required to obtain reasonable accuracy and report a thorough scientific evaluation of your model. Mention any challenges you faced in trying to train this model and explain how you overcame them.

Task 2 - Generation

In the real world, CAPTCHAs are unpredictable and do not belong to 100 easy classes. In this subtask, you will improve upon your architecture to extract the text itself present in the image i.e. we input the image, and the output is the text embedded in the image. Keep in mind that words can be of variable length, which you will need to account for. Be scientific in your evaluation. This task will require moderate tinkering with architectures and hyperparameters to achieve reasonable performance — document everything you've done. *We do not expect you to solve this task entirely but we do expect meaningful forward progress.*

Task 3 - Bonus

In the task, we will work exclusively on the bonus set. Generation becomes harder in this setting since the model now needs to learn how to output both in the forward direction and in reverse. This makes training slightly more challenging, but we're sure you can figure it out!

Pointers

- We strongly prefer that you do this task in PyTorch. This is not just our preference but also the research standard.
- Please experiment and save all experiments you conduct. If you make any interesting or surprising inferences, make sure to mention them.
- You may use Colab or Kaggle for access to GPUs; free-tier compute is sufficient for this task.
- All the solutions are expected to be a neural network trained from scratch. No OCR libraries.

Paper Reading Task

Learning Transferable Visual Models From Natural Language Supervision
[arxiv.org/abs/2103.00020]

For any doubts regarding this task please directly email - sreeram.vennam@students.iiit.ac.in, with cc to debangan.mishra@students.iiit.ac.in.