Simplifying and "Breaking" Bad Captchas via GANs

Nirvan Patil | BITS Pilani, Goa Campus

OVERVIEW

1. Introduction & Motivation
2. Dataset Generation
3. Classification
4. Simplification via GAN(s)
5. Results
6. Limitations
7. Future Work

TASKS COMPLETED

1. Dataset Generation
   a. Easy, Medium & Hard
   b. Bonus Data

2. Classification
   a. 100 Class Custom Dataset
   b. Comparison between ResNet, Simple CNN, ZS CLIP and Linear Probe CLIP
   c. # Samples, Efficiency, and Performance Tradeoffs

3. Generation
   a. Pix2Pix and Cycle-GAN for "simplifying" CAPTCHAs
   b. OpenCV automated pipeline for solving easy CAPTCHAs
   c. Bonus Task: Reverse Generation

Introduction & Motivation

Introduction

- ***What is OCR?***
  - Optical Character Recognition (OCR) is the process of converting images of typed, handwritten, or printed text into machine-encoded text.

- **CAPTCHA: A brief history**
  - Early implementations of CAPTCHA [1] (Completely Automated Public Turing test to tell Computers and Humans Apart) were text-based, while modern solutions (re-CAPTCHA [2]) improve robustness by making users identify everyday objects in images.
  - This project's focus remains on “solving” the former, older CAPTCHAs.

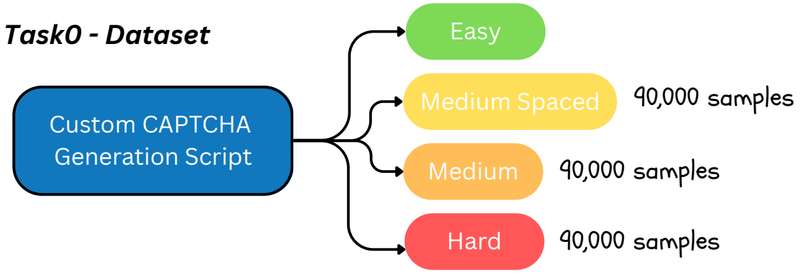
- ***Existing Solutions***
  - CAPTCHA remains a legacy problem [3, 4, 5], which has only become easier with the rise of Modern Computer Vision solutions.
  - In recent times, multimodality via Vision Language Models [1, 2, 3] has trivialised the task of traditional text-based CAPTCHAs.

Motivation / Abstract

- ***Moving away from SOTA***
  - While Vision-Language Models like CRNNs [5], CLIP [6], LLaVA [7] and traditional OCR libraries [3] offer effective solutions for solving text-based CAPTCHAs, this study aims to explore an alternative, less-explored approach.
  - Although [9] highlights the need for a modern General OCR Theory (GOT) leveraging Vision-Language Models (VLMs) to meet the increasing demand for intelligent interpretation of man-made optical characters, they acknowledge that OCR 1.0 (legacy solutions) remains sufficient for task-specific use cases, i.e. CAPTCHA.
  - This statement compounds the aim of the study —To explore the simplest of OCR 1.0 techniques while potentially improving them via “visual-simplification” through Generative Adversarial Networks (GANs) [10, 11, 12]
  - This study builds on the approach proposed in [13], but diverges by specifically exploring the effectiveness of different mappings from CAPTCHA to basic plain text images.

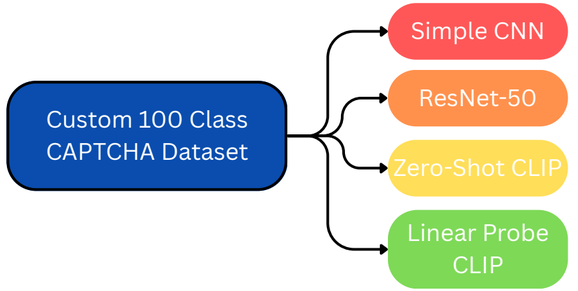
Pipeline Overview

Task0 - Dataset



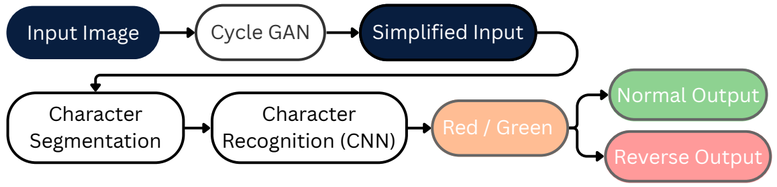
3 to 7 character long CAPTCHA sequences

Task1 - Classification



Trained across varying dataset lengths

Task2 - Generation + Task3 Bonus



[1] CAPTCHA, [2] re-CAPTCHA, [3] Tesseract, [4] OCROpus, [5] CRNN, [6] CLIP, [7] LLaVA, [8] GPT-4V, [9] GOT OCR 2.0, [10] GAN, [11] Pix2Pix, [12] CycleGAN, [13] A GAN based CAPTCHA Approach

Dataset Generation

Motivation

A robust dataset equates to a robust Model

- ***Synthesising the Dataset***
  - As outlined required by the problem statement, CAPTCHA datasets across varying "difficulty" levels were generated (no external datasets used as instructed)
  - To simulate occlusions and distortions in real-world CAPTCHAs, multiple fonts, varying character colours, distortions (skew, rotation, lines, dots) and background noise were engineered.
  - Heavy inspiration was drawn from [14], building additional features on top of it.

Dataset!

- ***Structure***
  - Character Length: To ensure maximum diversity in the dataset, CAPTCHAs of length 3 to 6 were generated.
  - Data Distribution: Every dataset has an equal number of images per length, per class (unique CAPTCHA sequence) and per difficulty.
  - A custom PyTorch DataLoader aids in this "equal distribution" data loading.

Datasets

Attribute	Classification Dataset	Generation Dataset	Bonus Data
Train Images	15,000 (Easy, Med, Hard: 5,000 each)	180,000 (Med, Hard: 90,000 each)	5,000 (Hard only)
Test Images	3,000 (Easy, Med, Hard: 1,000 each)	—	500 (Hard only)
Total Classes	100 (unique CAPTCHA seq.)	1800 (unique CAPTCHA seq.)	1000
Images per Class	150 (150 shots)	180 (180 shots)	5
CAPTCHA Info	Lengths: 3, 4, 5, 6, 7 Images per Length: 3,000	Lengths: 3, 4, 5, 6 Images per Length: 45,000	Lengths: 3, 4, 5, 6 Images per Length: 1,250
Data Path	`task1_classification/data`	`task0_dataset/med + /hard`	`task0_dataset/data_bonus`

- ***Character Classification***
  - As will be described in later sections, a dataset was constructed to train a CNN to classify individual characters.
  - Total Train Images: 62 (A-Z, a-z, 0-9) x 50 shots per character = *3100*
  - Total Test Images: 62 x 10 = *620*
- ***Class Overlap***
  - *Other than the characters and classification dataset, all other datasets have zero overlapping classes between the train and test splits.*

Docs

GitHub

Each task folder contains a "dataset.py" or similar to generate the datasets.

Availability

Owing to Git storage limits, only the character dataset is available as .zip under task2_generation/data_chars/data_chars.zip

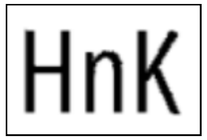
Directory Structure

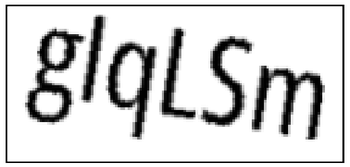
Although images cannot be shared, the hierarchical folder structure of each dataset can be viewed under their respective task folders on GitHub.

[14] GitHub LepTrue/Captcha/src/captcha/image.py

Dataset Examples

Easy Difficulty





Random Rotation Data Augmentation applied as otherwise each sample of a given class would be identicle

Medium Spaced

Medium Difficulty + Space between each character. This represents the target transformation (simplification) for the GAN

Medium

Medium Difficulty: As described in the table on the previous slide.

Hard

Hhard Difficulty: As described in the table on the previous slide.

Bonus

Green Background

Red Background

Bonus: Hard + Green/Red tinted background.

Methodology Classification

Motivation

- ***Building Blocks***
  - Classification, the task of assigning a label to an entire image based on its visual content, remains a strong test for the feature learning capabilities of a model.
  - While CAPTCHA and OCR tasks do not fit into the classification regime owing to infinitely many possible classes, limiting samples to 100 classes remains an effective way to study:
    - CNNs' Efficacy in CAPTCHA Representation
    - Easy to "learn" versus Emergent Features
    - Efficacy of Natural Language Supervision (CLIP)

- ***Designing the Study***
  - In line with the problem statement, a CAPTCHA dataset of 100 classes (where each class corresponds to a unique capitalisation-sensitive CAPTCHA sequence) was created (Ref. Prev Slides.)
  - To analyse the performance and representation learning trends of CNNs and CLIP, the dataset is further stratified based on two factors:
    - CAPTCHA difficulty levels — Easy, Medium, and Hard (equally distributed)
    - CAPTCHA lengths — 3, 4, 5, 6, and 7 characters

Comparisons

- ***CAPTCHA Representations***
  - End-to-end fine-tuning well captures the representation learning ability of a model [6].
    - Models are trained on varying datasets, each (train & test) containing the same 100 classes.
  - We observe: (Ref. Tab. 1 [Right])
    - Expected increase in Medium and Hard Test Accuracy with increased dataset size.
    - SimpleCNN outperforms ResNet-50 at larger dataset sizes, suggesting that, as noted in [6], ImageNet models remain overfitted to their pretraining data.
    - Linear Probe clip, despite having 1/80th trainable parameters as the SimpleCNN, remains a competitor.

- ***Emergent Features***
  - All models find it difficult to generalise to Hard samples, which only differ from the Medium ones by the addition of Occluding Lines and Dots
    - This showcases that such patterns retain an emergent complexity rather than a smooth trend, requiring a large architecture (CLIP retains >50% Hard Acc @1000 samples) or explicit training (Training on Hard samples).
  - Interestingly, ResNet-50 consistently underperforms on the Hard set compared to the SimpleCNN, despite having 5× more trainable parameters.
    - Since only the last two blocks of ResNet-50 are retrained, this suggests that effective CAPTCHA representation depends heavily on earlier CNN layers.

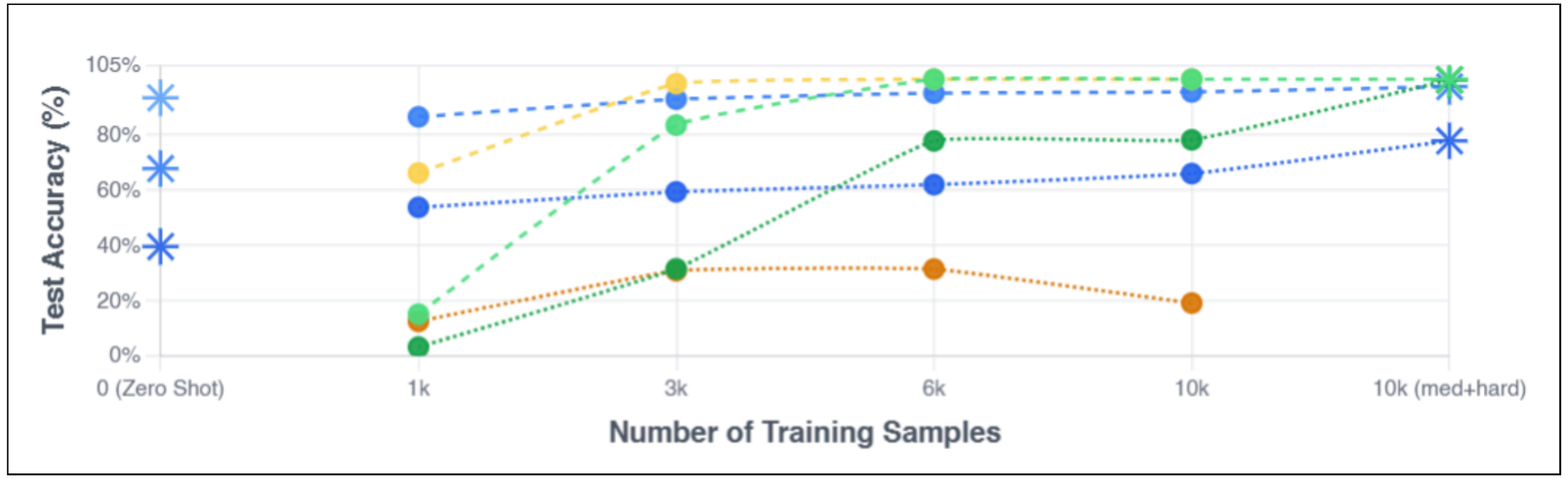
# Train Samples	Train Acc	Medium Acc	Hard Acc
SimpleCNN (Conv Layers: [64, 128, 256, 512, 512], Params: 3.95M)			
1000	100	15.1	3.2
3000	100	83.4	31.5
6000	100	100	77.7
10000	100	100	78.1
10000 (med+hard)	100	100	99.7
ResNet-50 (Pre-Trained ImageNet, Params (Layer 3+4): 22.26M)			
1000	100	66.1	12.5
3000	100	98.4	30.8
6000	100	100	31.4
10000	100	100	19.1
Linear Probe CLIP ViT-B/32 (Params: 51.3K)			
1000	100	86.4	53.7
3000	100	92.8	59.3
6000	100	95.0	61.9
10000	100	95.4	65.9
10000 (med+hard)	100	97.3	77.7

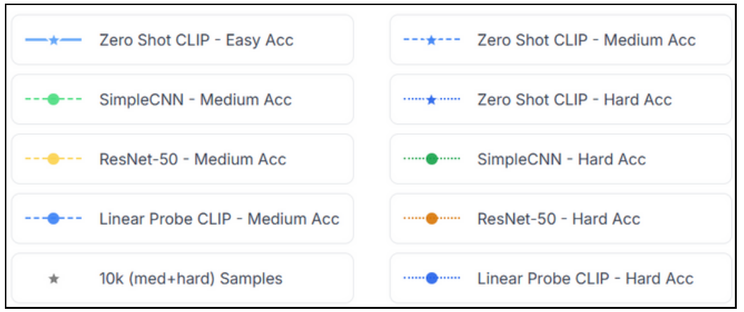
Tab 1: Classification Accuracy across varying models and dataset sizes
(Unless specified, Train split consists of Equal Easy + Medium samples)

[6] CLIP

Results Classification

Detailed Comparisons





- ***Overfitting of ResNet-50***
  - A noticeable drop in Accuracy on the Hard Test split is observed as we increase training data from 6k to 10k samples.
  - This alludes to ResNet-50 heavily overfitting the training data, failing to generalise to the additional noise in Harder images.

CLIP and Prompt Engg.

Prompt Format	Easy (1000)	Medium (1000)	Hard (1000)
`"{label}"`	89.7%	66.2%	35.9%
`The word "{label}"`	93.3%	70.8%	39.5%
`The word "{label}" in the image`	93.2%	67.7%	35.1%

Table 2: Comparing the efficacy of 3 levels of prompt engineer on Zero-Shot CLIP

- ***Natural Language Supervision***
  - CLIP [6] demonstrates that zero-shot performance serves as a strong indicator of task adaptability, emphasising the importance of prompt engineering for optimal results.
    - Specifically, CLIP performs better on OCR tasks when the target CAPTCHA or text is enclosed in double quotes (“ ”).
  - However, the impact of additional layers of prompt engineering beyond this initial optimisation remains unclear.
    - Table 2 [Left] shows that excessive prompt complexity (Row 3) can actually degrade zero-shot transfer performance, suggesting there is an optimal level of prompt sophistication.

[6] CLIP

Methodology Generation

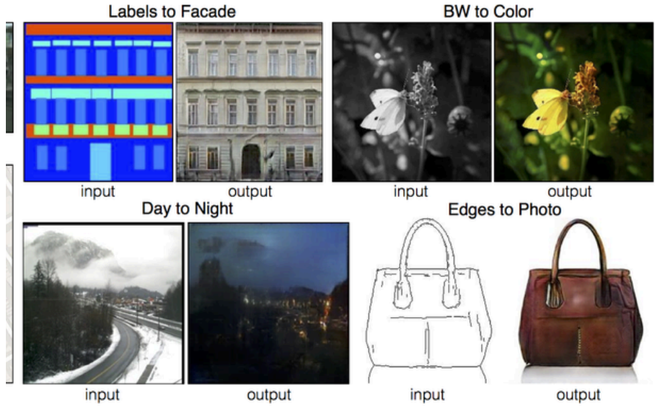
Motivation

- ***Exploring the Unexplored***
  - As elaborated in the Motivation section, legacy as well as modern solutions render text-based CAPTCHAs as a “solved problem”.
  - As a result, there remains motivation to push away from fine-tuning tried and tested architectures and explore an alternative tangent.
  - This inspiration is drawn from [13], which presents a GAN-based solution to “simplify” CAPTCHAs by removing “security features” (occluding lines, noise patterns, etc.)

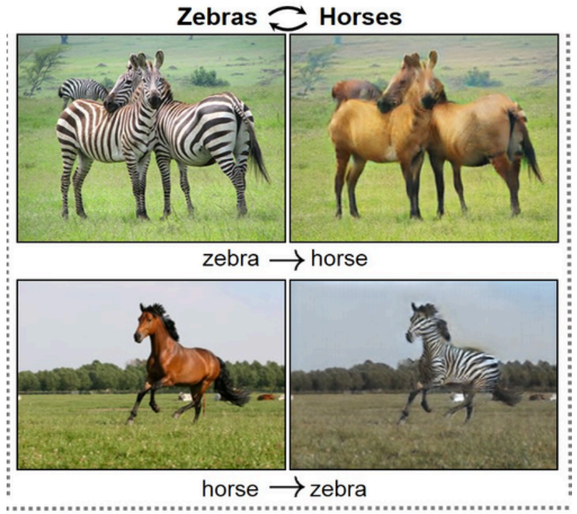
- ***Goal of the Study***
  - CAPTCHAs traditionally resist automated reading by introducing occluding strokes, speckles, and random artefacts that frustrate simple segmentation.
  - If each character could instead be isolated, the recognition task collapses to a fixed 62-way classifier (A–Z, a–z, 0–9), which also simplifies the bonus task to a background detection problem rather than a “reverse generation problem”.
  - As a potential solution, this study aims to explore image-translation GANs, specifically Pix2Pix and CycleGAN, to learn a “denoising” transform from noisy CAPTCHAs to their idealised, security-feature-free versions, while quantifying accuracy gains on utilising these synthetically mapped clean CAPTCHAs versus their original “noisy” counterpart.

Pix2Pix and CycleGAN

- ***Overview***
  - Pix2Pix and CycleGAN are specialised instances of Generative Adversarial Networks (GANs), where a generator learns to produce images and a discriminator learns to distinguish real from generated images.
- ***Pix2Pix***
  - A conditional GAN (cGAN) designed for paired image-to-image translation tasks, learning a mapping given aligned input–output image pairs.
  - Pix2Pix’s generator uses a U-Net encoder–decoder with skip connections, enabling fine detail preservation by directly copying feature maps from encoder layers into matching decoder layers.
  - Its discriminator is a “PatchGAN,” classifying patches rather than the whole image, which focuses learning on high-frequency structure and dramatically reduces parameters
- ***CycleGAN***
  - An unpaired translator that tackles image-to-image mapping without paired data by introducing cycle consistency losses to enforce rich feature representations
  - CycleGAN employs two generators and two discriminators, with each discriminator judged only on its target domain.
  - The cycle consistency loss in CycleGAN ensures that translating an image to the target domain and back reconstructs the original, ensuring the model learns relevant and rich feature representations



Pix2Pix Example Usecases



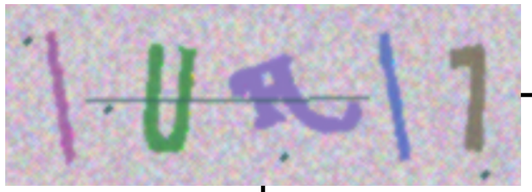
CycleGAN Example Usescases

[10] GAN, [11] Pix2Pix, [12] CycleGAN, [13] A GAN based CAPTCHA Approach

Methodology Generation

Ideal Pipeline

Input Noisy CAPTCHA
Label: IUHI7



CycleGAN / Pix2Pix

Image-to-Image Translation

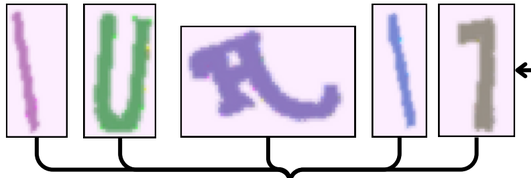
Clean CPATCHA image
Post-GAN transformation

Lines, dots and noise removed. Space between characters added. Rotation, colour and font of each character maintained

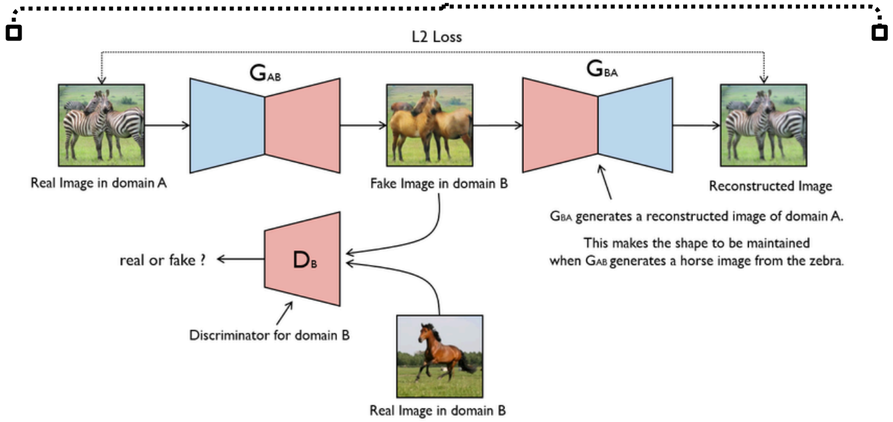
OpenCV Character Segmentation

Binary Masking

OpenCV identifies X limits of each character. Y limit of each character determined by the maximum "y" value of a non-zero pixel in corresponding binary mask. This prevents characters like "i" holding two bounding boxes, one for the stem and one for the dot.



CNN for Character Classification



E.g.: CycleGAN inner working

Background detection

Green/Red

Red

+

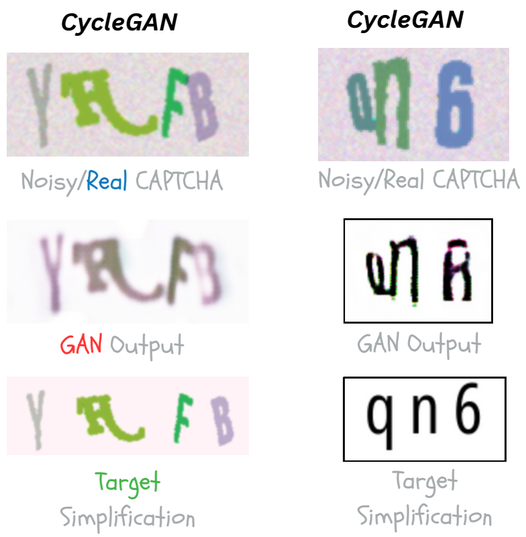
Reversed Text:
7IHUI

Results Generation

Image to Image Translation

- ***Difficulties***
  - CycleGAN/Pix2Pix was trained from scratch using an aligned dataset, where each pair consisted of a noisy (real) CAPTCHA image from Domain A and its corresponding clean (simplified) version from Domain B.
  - Despite extensive hyperparameter tuning, utilising large datasets (90,000 Train Images), given the constraint of training a model from scratch, both Pix2Pix and CycleGAN did not produce sufficiently crisp results to enable the Character Segmentation Module. (Ref. Fig 1. Right)

- ***CycleGAN versus Pix2Pix***
  - Although Pix2Pix is aimed at paired Image-to-Image translation, it performs poorly as compared to CycleGAN, which, despite not being aimed at paired translations, supports this option.
    - This is likely due to CycleGAN's novel Cycle Consistency loss ensuring rich mapping from both noisy CAPTCHAs → clean CAPTCHAs and vice-versa, ensuring rich feature representations are learnt
  - Soft, misshapen and interestingly misplaced order of characters leaves room to be desired before the simplification pipeline can produce meaningful results.
  - Potential solutions discussed in the Limitations Section.



- ***Different Degrees of Simplification***
  - Mapping to basic plain text (right) remained exceedingly challenging owing to the drastic domain shift
  - While a translation to a similar character font and rotation (left) fared better, signifying a lower domain shift.
    - Although (left) shows promising potential results, convergence and stability in outputs remained difficult.
- ***A path forward***
  - This comparison showcases that adjusting the degree of domain shift can greatly boost performance.
  - This strengthens motivation to explore a domain shift just enough to enable effective Character Segmentation while being easy enough for the GAN to translate.

Character Segmentation

- ***Conditional Strong Performance***
  - The Character Segmentation pipeline (Ref. Prev. Slide) involving individual character extraction followed by a CNN for classification represents a fast, efficient and light-weight solution, which conditionally showcases strong results.
  - The caveat remains that character overlap and occluding noise must remain absent.
    - The aim of the GAN module is exactly this: to remove occlusions and noise; however, currently, there remains progress to be made along this direction.

	Medium Spaced (90k)	Medium (90k)	Hard (90k)
Character Segmentation Accuracy	96.87%	40.5%	10.2%

Table 3: Character Segmentation Accuracy across difficulties

Bonus Generation

- In an idealistic scenario, where the GAN is able to successfully "simplify" the CAPTCHA, the Bonus Generation Module leverages individual characters obtained from the Character Segmentation Module, trivialising reverse or forward output as a matter of which character is printed first.
- Every image correctly handled by the Character Segmentation Module (Ref. Tab 3), the Bonus Generation Module successfully performs its Task.

Limitations CLIP

Poor Image-to-Image Translation

- ***Addressing Current Performance***
  - While the study explores an alternate and exciting tangent by veering away from the current or legacy OCR SOTA, tangible performance leaves much to be desired.
  - CycleGAN, the stronger of the two Image-to-Image Translation models tested, showcases strong potential, being able to generate acceptable "translations" despite:
    - The models were trained from scratch (given the assignment guidelines)
    - Limited Data (access to only synthetically generated data) and Compute

- ***Potential Improvements***
  - CycleGAN exhibited a strong positive performance trend with an increase in the number of trainable parameters.
    - Although current computational limits remain a hurdle for this path, a large/powerful Generator and Discriminator are likely to greatly increase performance.
    - However, this remains at the cost of keeping the solution lightweight.
  - Additional improvements are listed under the "Future Work" section.

Lack of Scalability

- ***Limited to CAPTCHAs***
  - While a GAN-based "simplification" pipeline presents an intriguing research direction, it suffers from poor scalability.
  - Current evaluations reveal that the GAN struggles with longer CAPTCHAs, and extending the approach to full-page text, as opposed to short phrases or single CAPTCHAs, is guaranteed to lead to a significant drop in performance.
  - This method, even when refined, remains constrained to handling only single or, at best, a few-word CAPTCHA/phrases.

Limited Scope

- ***Not task adaptable***
  - The proposed pipeline remains exclusive to "solving" only CAPTCHAs, which remain a legacy utility being replaced rapidly by the more modern reCAPTCHA.
  - On the contrary, potential VLM/CRNN-based solutions trained for CAPTCHA identification may hold value in other multi-modal tasks.

Future Work

- ***Task-specific GAN***
  - CycleGAN and Pix2Pix remain foundational/general-purpose Image-to-Image translational models; extensive work is required to refine said architectures to ensure relevance for text-containing images, where edge clarity and per-pixel accuracy hold greater value
  - Adopting/utilising NVIDIA's Pix2PixHD [15] is a promising direction.

- ***Optimum "Simplification"***
  - While current results do not achieve acceptable "simplification" of CAPTCHAs, varying the degree of domain shift required between "noisy" and "clean" images (Ref. Prev. Slide) can greatly boost results.
  - Thus, finding an optimal "degree of simplification" remains essential to ensure both the GAN and Character Segmentation Modules work harmoniously.

- ***Modified Loss Function***
  - Incorporating a loss component that aligns with the feature representations of a CAPTCHA-pretrained CRNN or VLM can effectively guide early convergence and enhance edge refinement during GAN training.

[15] Pix2PixHD

CONCLUSION

This study investigates the evolving landscape of CAPTCHA (Completely Automated Public Turing test to tell Computers and Humans Apart) through both classification and generation perspectives. While the classification component may have limited direct real-world application, the experiments reveal that occlusions such as lines, dots, and background noise introduce emergent complexities that challenge traditional models. These challenges often necessitate either significantly larger training datasets or the use of powerful pre-trained architectures to achieve effective feature representation, particularly in the early layers of CNNs. On the generation front, although current efforts diverge from state-of-the-art approaches and do not yet yield robust results, the novel direction supported by promising initial outcomes using CycleGAN suggests a compelling trajectory for future exploration, as discussed in the Future Work section.

BITS Pilani
K K Birla Goa Campus

THANK YOU