Synthetic data generation using GANs

(Generative Adversarial Neural Networks)

Varun Singh

Target Problem

Generate this:

+ “248 FH8”

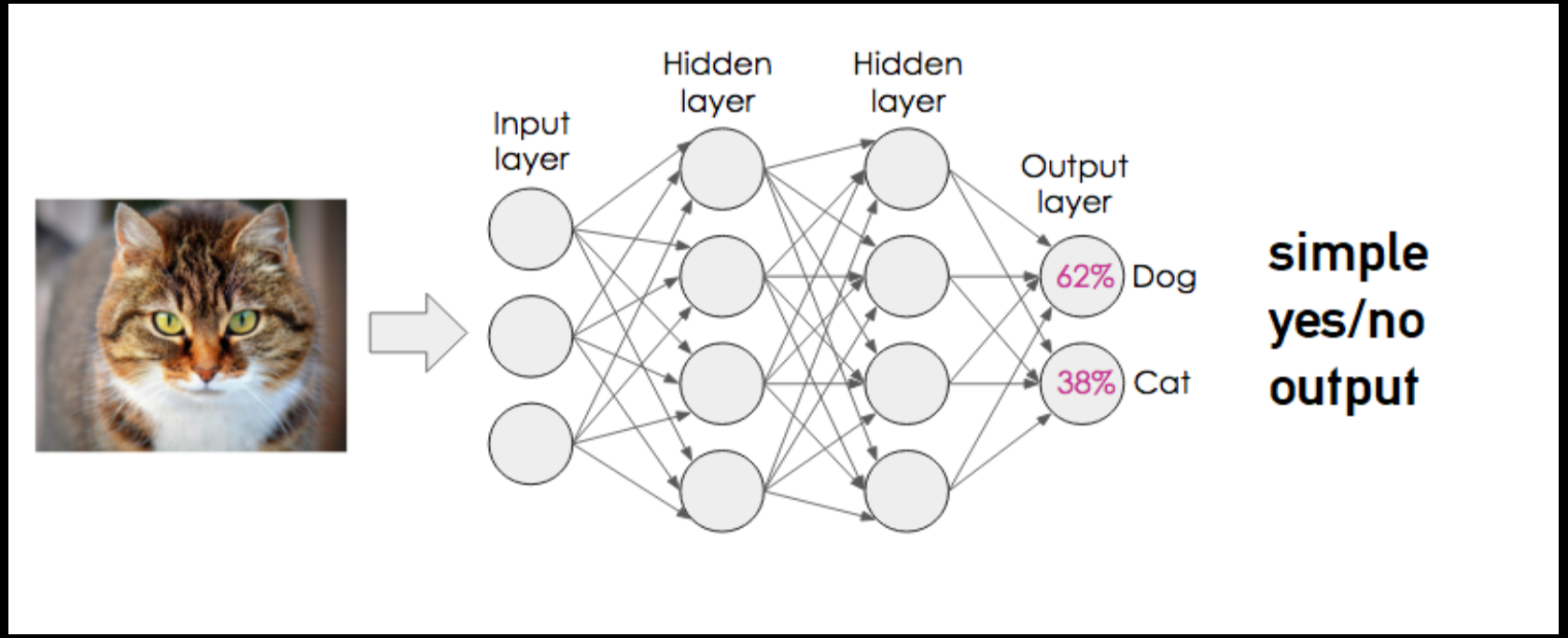
Or at least this:

+ “MH12 GR 9168”

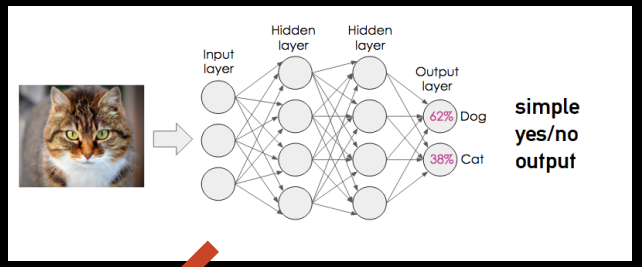
We want use this data to train ANPR programs.

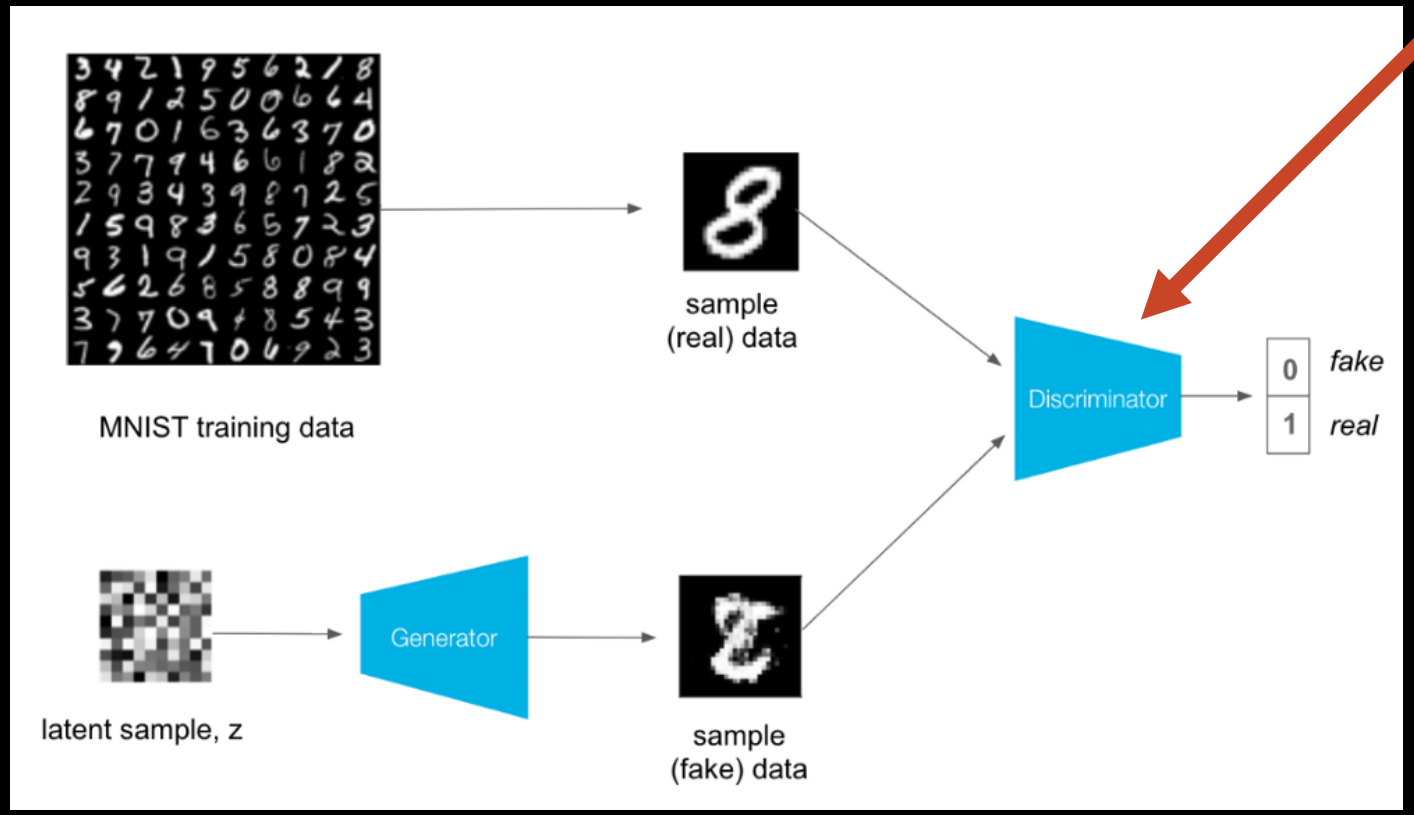
What are GANs? Not this:

(This is just a discriminator. INPUT → CAT/DOG)



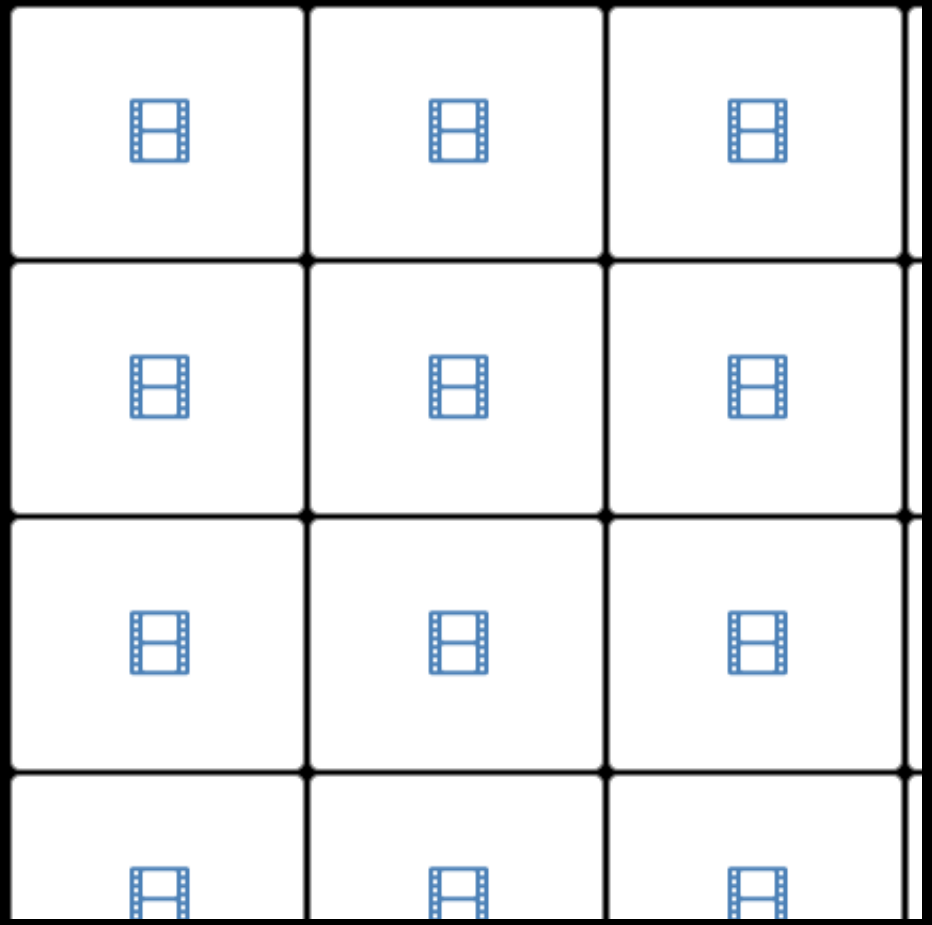
A simple GAN.



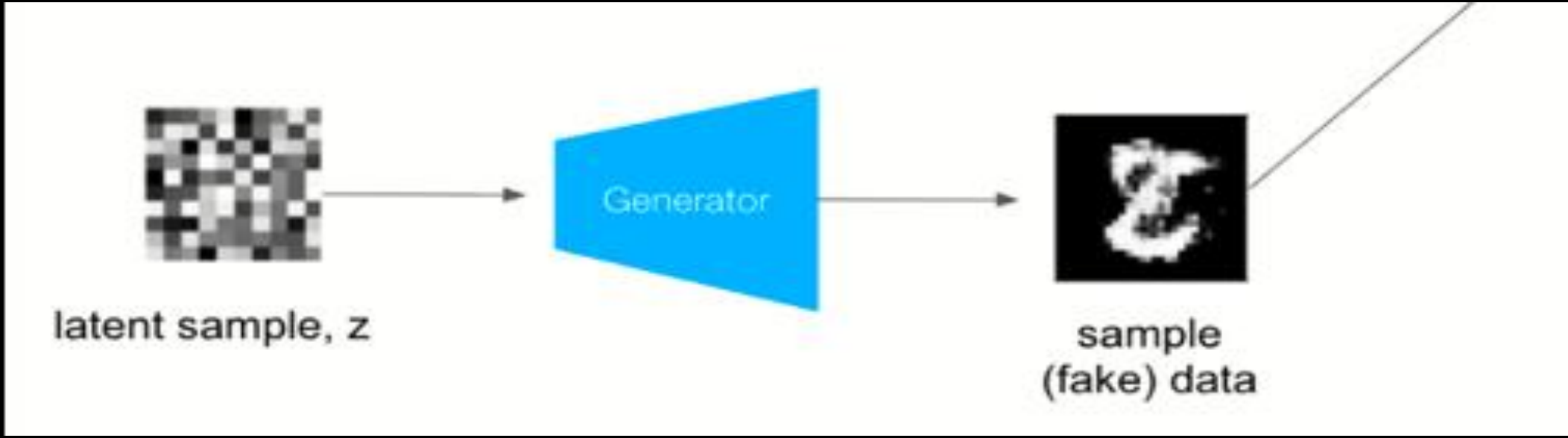


Both neural networks evolve in tandem. Discriminator tries to tell real data from fake. Generator must train to produce convincing images to fool the discriminator.

Results of a simple generative network.



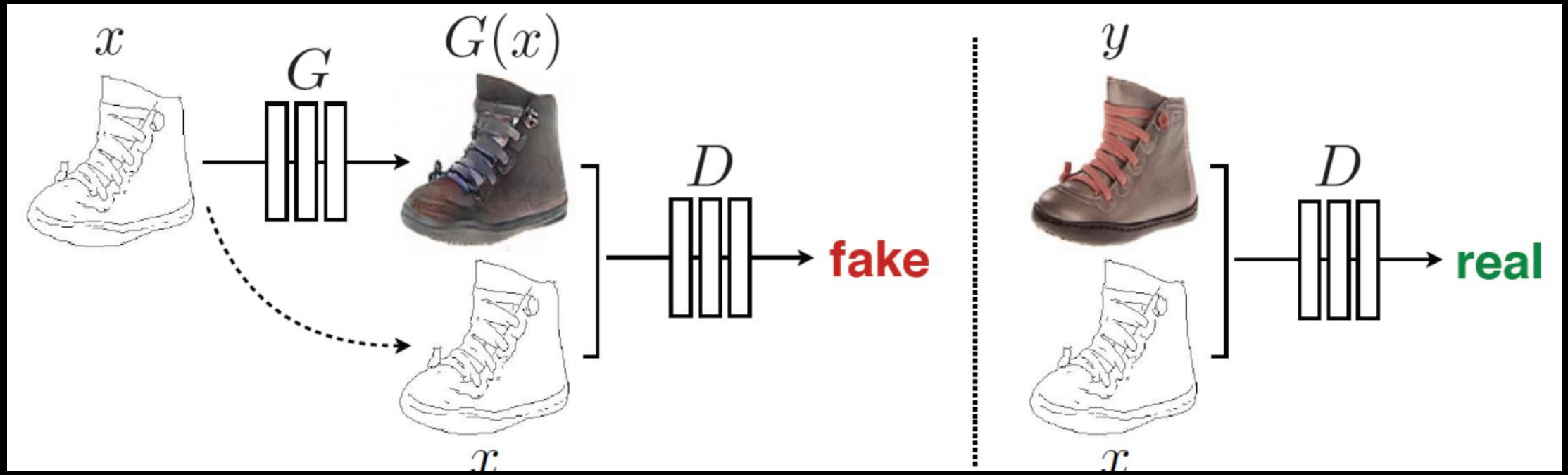
Problems with such a GAN



- RANDOM INPUT → ANY NUMBER (G is a blackbox.)
- Doesn’t supply labelled data.
- Cannot produce annotated license plate images.

Conditional GANs (Pix2Pix in particular)

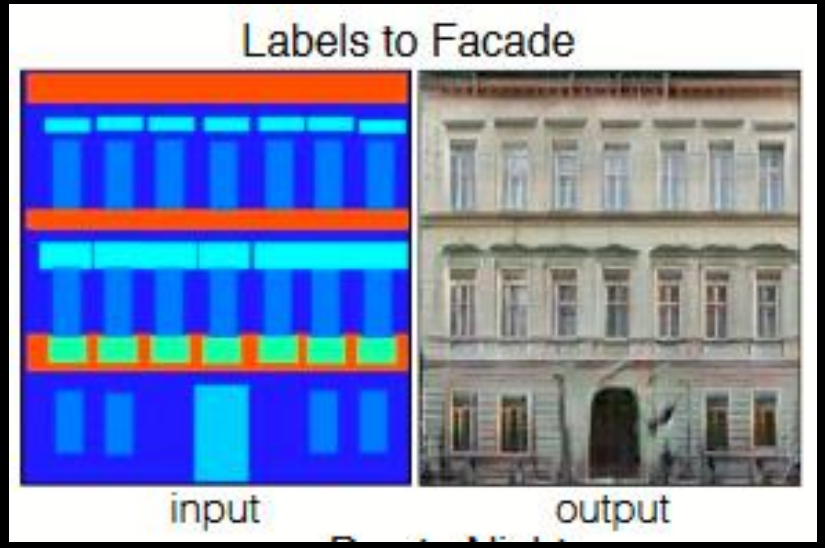
We're working with labelled pairs of data. Doing Image to Image translation. Call it a fancy coloring app.



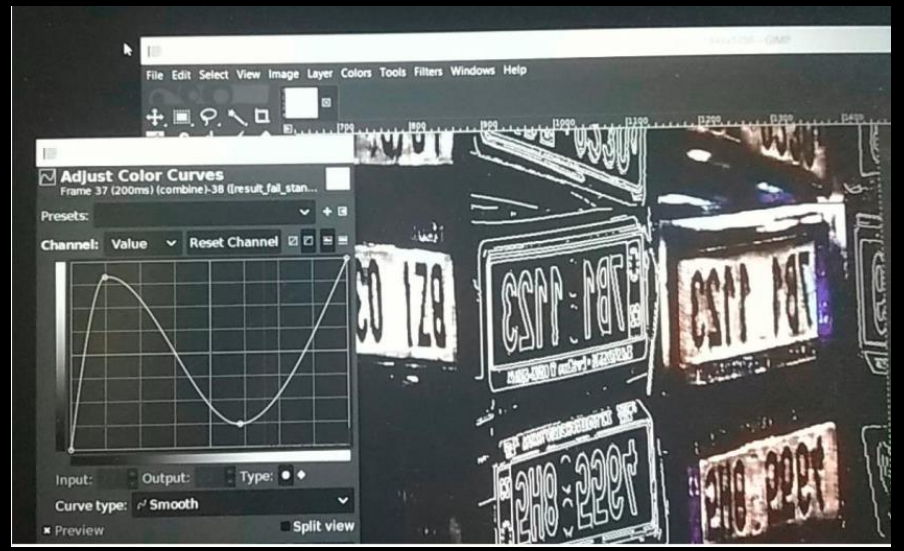
y = G(x) will have same label as x.

Excerpt from pix2pix paper

Architectural labels→photo 400 training images from [45], trained for 200 epochs, batch size 1, with random jitter and mirroring. Data were split into train and test randomly.



Low data, little training: Poor results

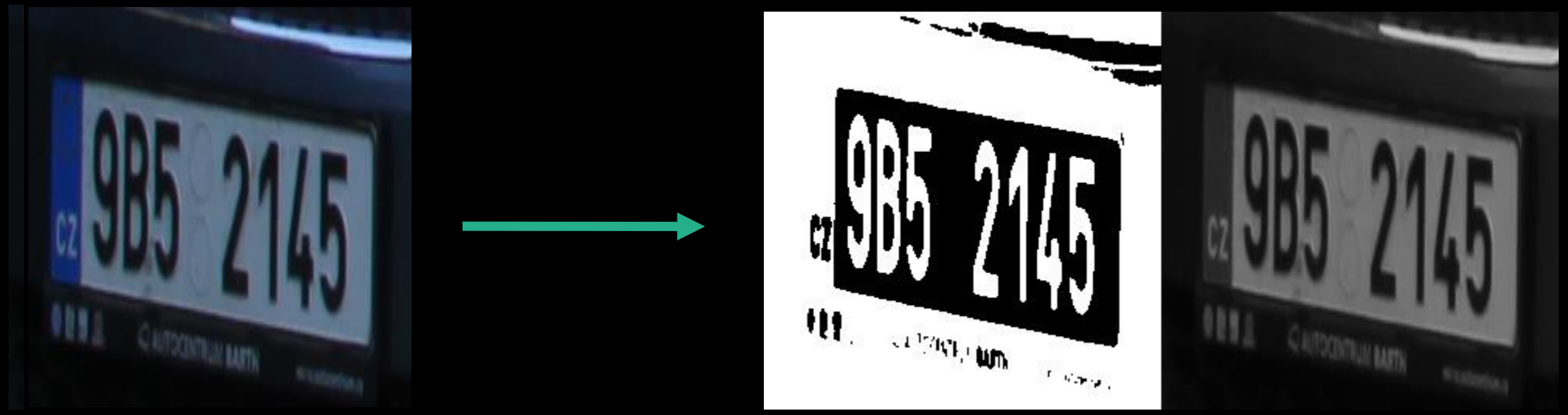


Color → Photograph FAILURE
Outline → Photograph FAILURE

Even if I brightened the generator output there was too much chromatic aberration.

SOLUTION: RGB → HSL →

data preprocessing



PHOTOGRAPH

SAMPLE INPUT PAIR
(2_bit → desired_greyscale_output)

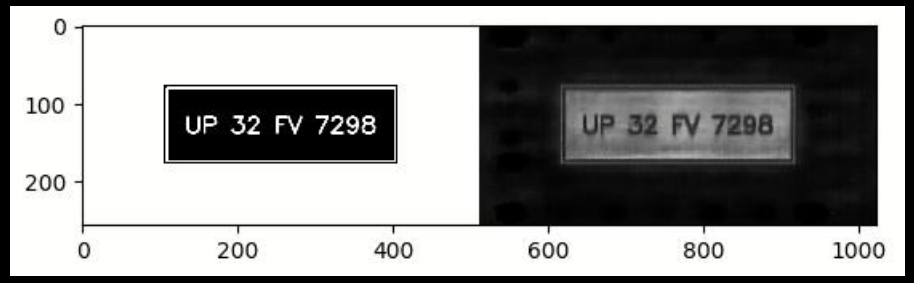
Drop the pesky, confusing RGB; change input image space to simple pure BLACK/WHITE and output to grayscale photograph.

Acceptable results

240 images, 10 epochs

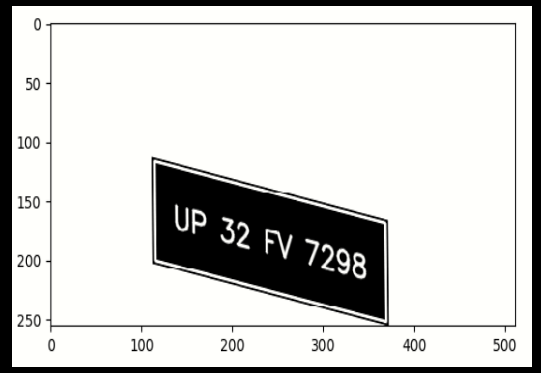
240 images, 200 epochs

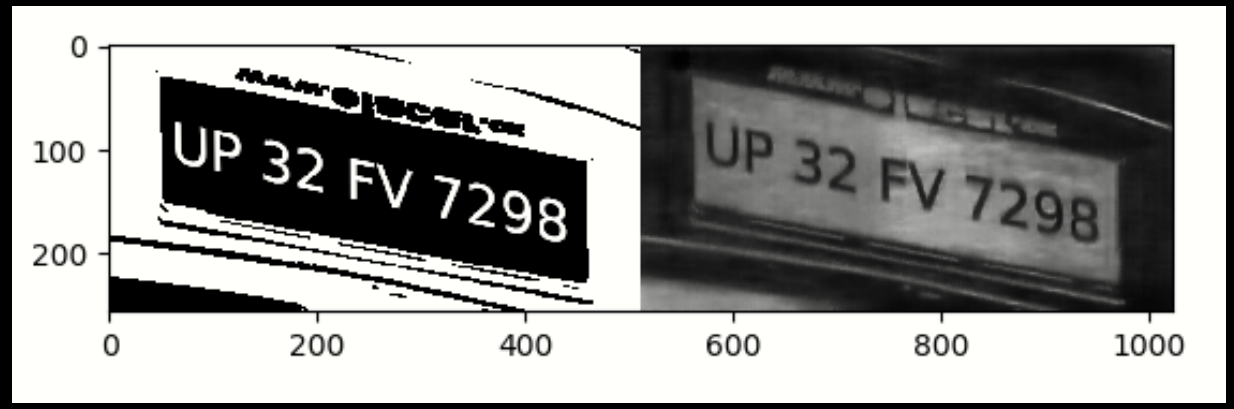
Generating arbitrary numbers



I could just do this, but the GAN has learned a lot more than simple shading.

Did a perspective transform and put number on a image model already trained on for better results.





Limitations of this approach

- That perspective transform matrix was calculated manually. Automating didn’t work out.
- No font variation
- It’s a shading AI.

Preparing templates by hand is unacceptable.

What’s possible?

- Models like DCGAN could generate photorealistic plate images from random noise input. It will thus be unlabeled.
- Need A LOT of diverse training data – which we don’t have. The raison d'être of my problem statement.

What's not possible?

- If we're trying to generate lots of synthetic data from a few images, then it is unrealistic to expect a model that can produce a picture from thin air (aka noise).
- Any viable solution will take structured, known input and morph it with some randomization.

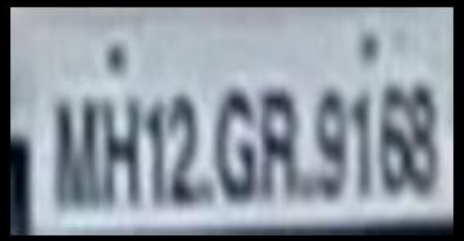
x 250 images

cannot be done

Reduction in scope

+ “248 FH8”

+ “UP 32 FV 7298”



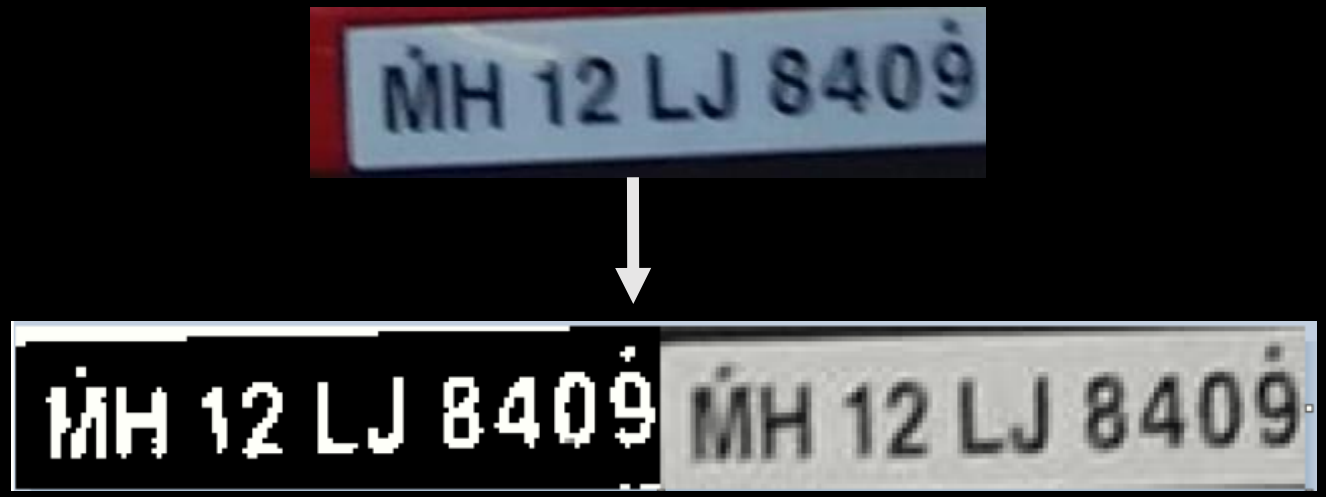
+ “MH12 GR 9168”

CelebA dataset has 2,00,000 annotated images. I don’t have anything like that. Deemed impossible.

I tried. Okay-ish results. Preparing a set of diverse templates is far too much manual labour. Hard.

Lower resolution. No background. Got enough data to mimic. Easy.

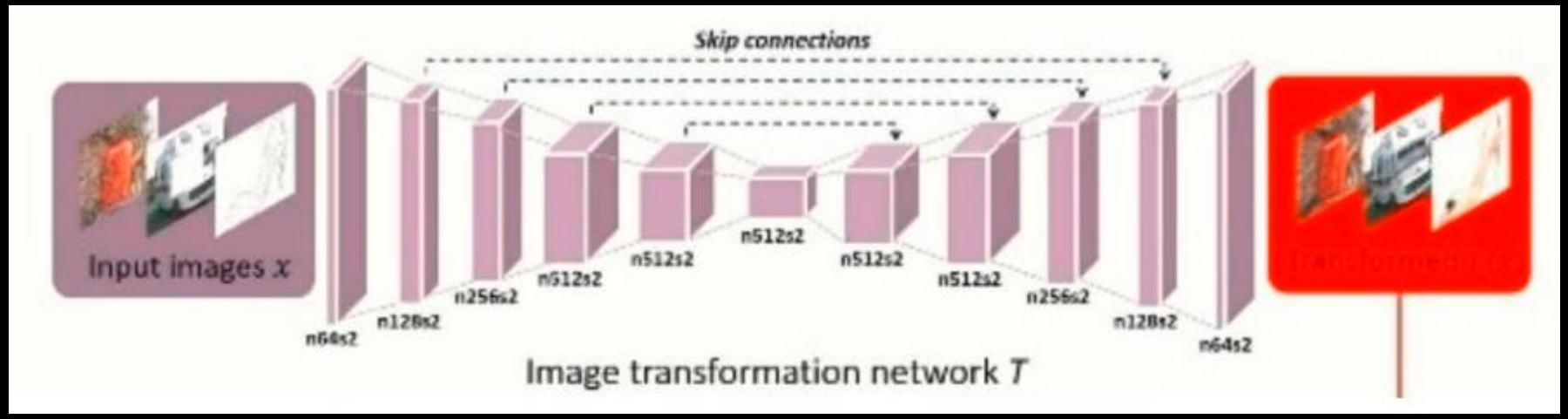
Data preprocessing



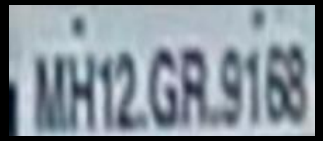
Sample Input Pair (2_bit → desired_greyscale_output)

- Snag: Too low resolution. Down-sampled to practically nothing in pix2pix.

Modifying generator architecture



128x32 image

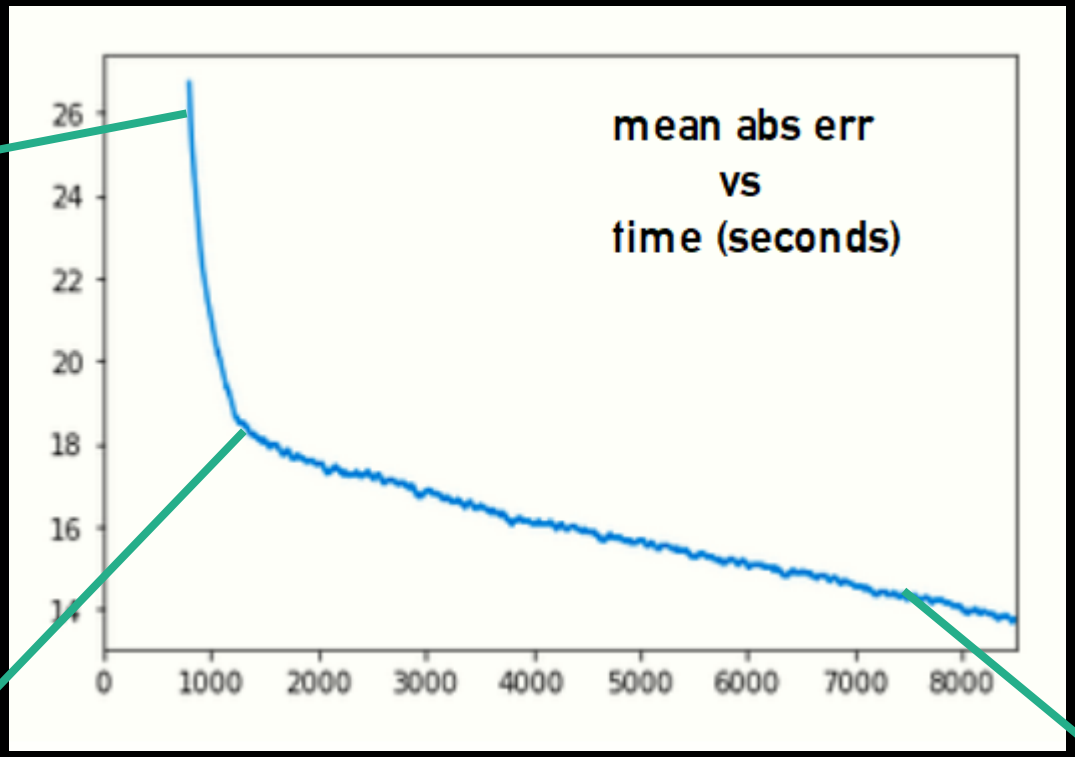


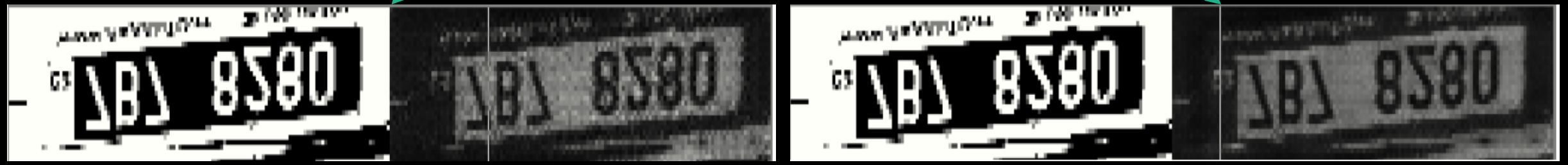
destructive down-sampling → 1 pixel → nonsense reconstruction

Solution: Reduce network depth. Use a ‘tiny-pix2pix’.

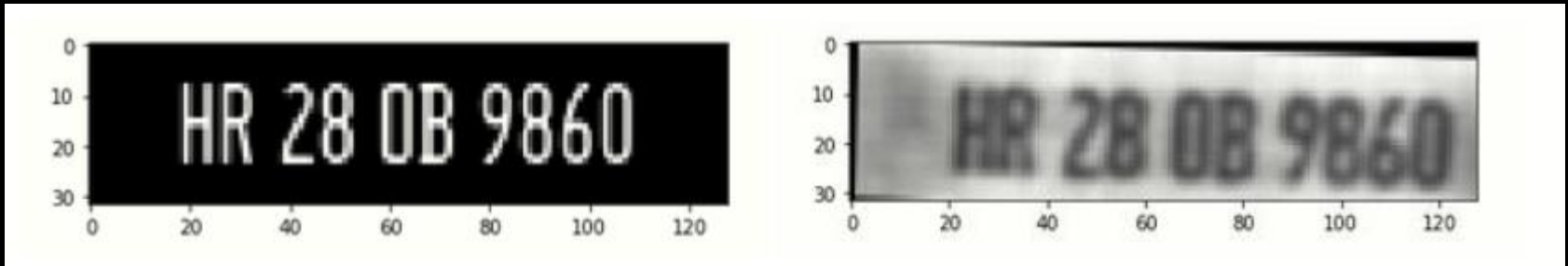
Training

Noise





Arbitrary Number Generation



Rules for Indian Vehicle Number Plates from Wikipedia.

↓

Generate random string

↓

Make input image

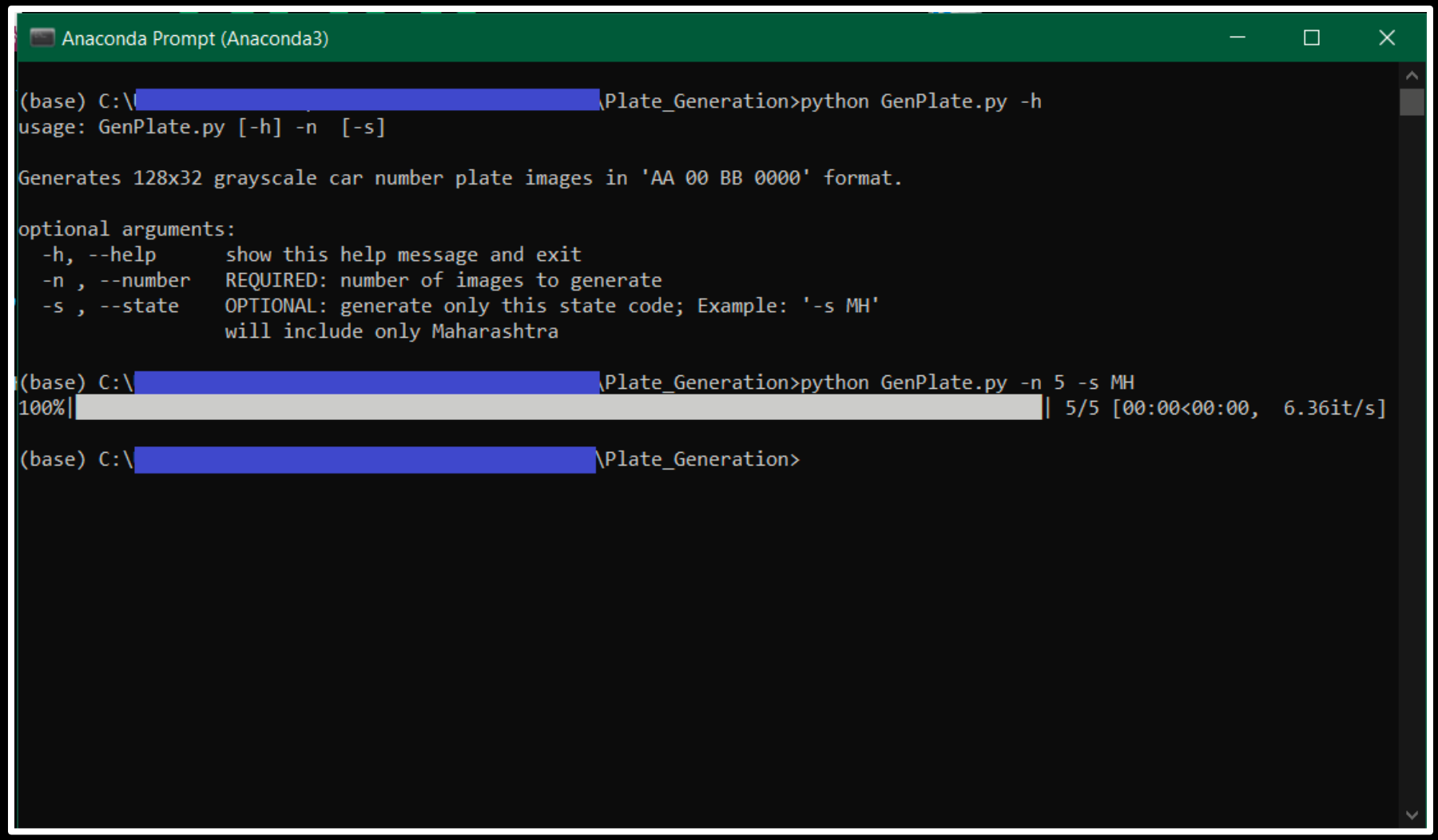
↓

Shade with GAN

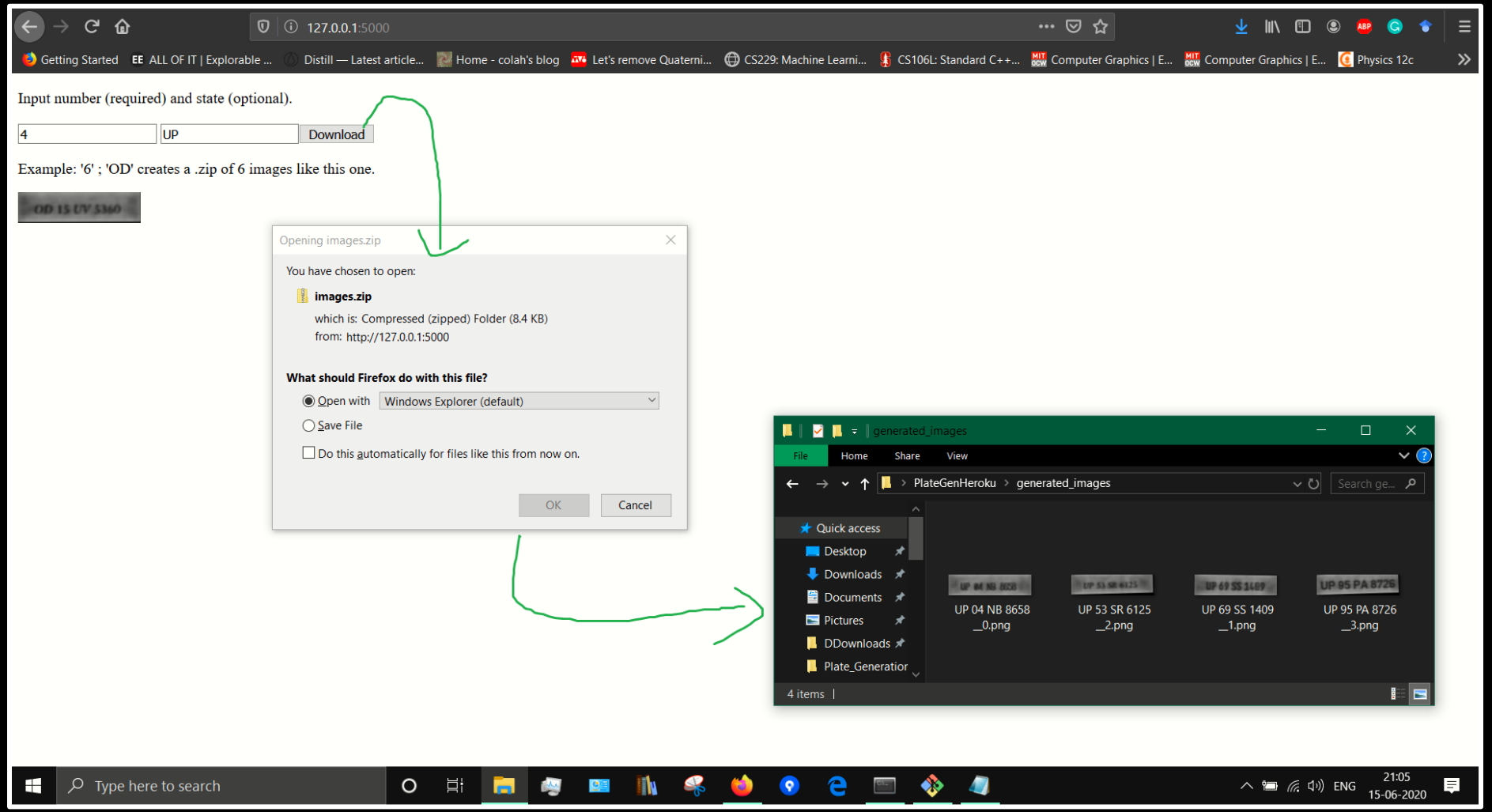
↓

Apply random jitter

Command line application



Web front-end



Extra: Correcting a blunder

```
86              u = image_pairs.cpu().numpy().astype(np.float32)
87              u = np.rint(((u + 1)/2)*255).astype(np.uint8)
```

- Forgot to change floats [-1,1] to integers [0, 255] before display and thus abandoned colored shading.
- Still unviable (because templates reqd.) but I should’ve caught this earlier.

Questions?