**Good afternoon everyone,**

I am Pankit Bhat from CDAC Pune and today I am going to present how we have used PARAM Shavak for solving automatic text based captchas using GAN and CNN which fall under the category of deep learning.

Deep learning models, in simple words, are large and deep artificial neural nets. A neural network (“NN”) can be well presented in a [directed acyclic graph](https://en.wikipedia.org/wiki/Directed_acyclic_graph): the input layer takes in signal vectors; one or multiple hidden layers process the outputs of the previous layer. The initial concept of a neural network can be traced back to more than [half a century ago](https://cs.stanford.edu/people/eroberts/courses/soco/projects/neural-networks/History/history1.html). But why does it work now? Why do people start talking about them all of a sudden?

The reason is surprisingly simple:

* We have a lot **more data**.
* We have **much powerful computers**.

A large and deep neural network has many more layers + many more nodes in each layer, which results in exponentially many more parameters to tune. Without enough data, we cannot learn parameters efficiently. Without powerful computers, learning would be too slow and insufficient.

Problem Statement:

Captchas (an acronym for Completely Automated Public Turing Test to Tell Computers and Humans Apart) are a widely used security mechanism to identify non-human intruders. The recent captcha attacks are scheme-specific, making them unstable when the captcha security features change. Today I introduce a generic yet practical approach to solving a text based captcha (scheme specific). The work heavily relies on a new deep neural net architecture called generative adversarial networks and its sub flavour pix2pix.

[Generative adversarial network](https://arxiv.org/pdf/1406.2661.pdf), short for “GAN”, is a type of deep generative models. GAN is able to create new examples after learning through the real data. It consists of two models competing against each other in a zero-sum game framework.

This approach has 3 steps:

First we generate captcha using a GAN based captcha Generator.

Then we rely on Pix2Pix implementation to pre-process any given captcha to strip its security features.

Then the stripped captchas are used to train a base solver which is a Convolution Neural Network (CNN)

For any neural network, the training phase of the deep learning model is the most resource-intensive task

While training, a neural network takes in inputs, which are then processed in hidden layers using weights that are adjusted during training and the model then spits out a prediction. Weights are adjusted to find patterns in order to make better predictions.

Both these operations are essentially matrix multiplications. A simple matrix multiplication can be represented by the image here

In a neural network, first array is the input to the neural network, while the second array forms its weight.

Easy, right?

Yes, if your neural network has around 10, 100 or even 100,000 parameters. A computer would still be able to handle this in a matter of minutes, or even hours at the most.

But what if your neural network has [more than 10 billion parameters](https://www.popsci.com/science/article/2013-06/stanfords-artificial-neural-network-biggest-ever/)? It would take *years*to train this kind of systems employing the traditional approach. Your computer would probably give up before you’re even one-tenth of the way.

Deep Learning models can be trained faster by simply running all operations at the same time instead of one after the other.

You can achieve this by using a GPU to train your model.

A GPU (Graphics Processing Unit) is a specialized processor with dedicated memory that conventionally perform floating point operations required for rendering graphics

In other words, it is a single-chip processor used for extensive Graphical and Mathematical computations which frees up CPU cycles for other jobs.

The main difference between GPUs and CPUs is that GPUs devote proportionally more transistors to arithmetic logic units and fewer to caches and flow control as compared to CPUs.

While CPUs are mostly applicable for problems that require parsing through or interpreting complex logic in code, GPUs are designed to the dedicated graphical rendering workhorses of computer games, and which were later enhanced to accelerate other geometric calculations

A GPU is smaller than a CPU but tends to have more logical cores (arithmetic logic units or ALUs, control units and memory cache) than the latter.

If you consider a CPU as a Maserati, a GPU can be considered as a big truck.

The CPU (Maserati) can fetch small amounts of packages (2 -4 passengers) in the RAM quickly whereas a GPU (the truck) is slower but can fetch large amounts of memory (~40 passengers) in one turn.

**Traditional CPU computing**

A central processing unit (CPU) also referred to as CPU is the hardware within the computer that carries out the instructions of a computer program by performing the basic arithmetical, logical and input/output operations of the system

Fundamentally CPUs execute a set of sequentially stored instructions called a program. The performance of a CPU depends on two factors namely the clock rate and the instructions executed per clock cycle (IPC). These two factors constitute the instructions per second quantity(IPS)

Now over the years as more work is done by the processor it has led to more consumption power and increased the transmission delay.

Moore’s law states that the number of transistors in a chip double every two years. As a result of this law the number of transistors on a chip increases rapidly. With the increasing number of transistors, the transmission delay of each individual transistor adds up to a huge transmission delay in the chip as a whole. This results in increasing the response time of the CPU.

Latency – Time taken to process one unit of data

Bandwidth – No of units of data processed per unit time

In the efforts to increase the bandwidth, a GPU uses a collection of more less efficient processing units (analogous to the bus in the previous example) unlike the CPU which uses one highly efficient processor (analogous to the race car)

An example in argument of increasing bandwidth over latency for GPUs

It is tolerable if one pixel appears takes a little longer to appear on screen when more pixels can be produced faster simultaneously