

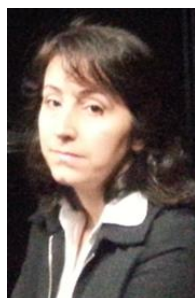


Inteligência artificial e aplicações em física

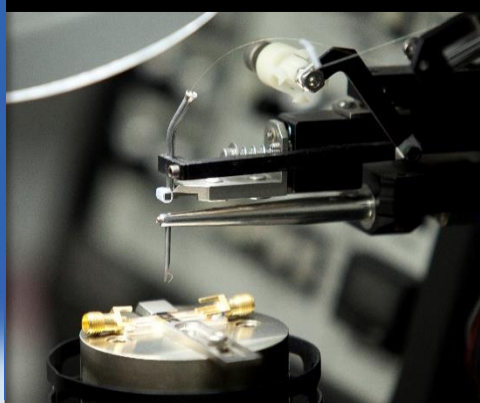
Clécio Roque De Bom – debom@cbpf.br



clearnightsrthebest.com



Elisangela L. Faria
(CBPF)

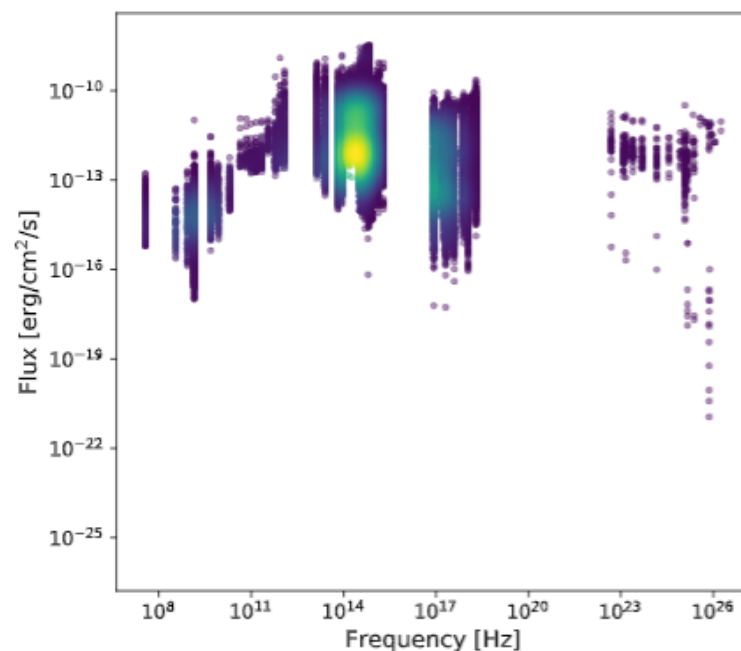


Blazars SED classification

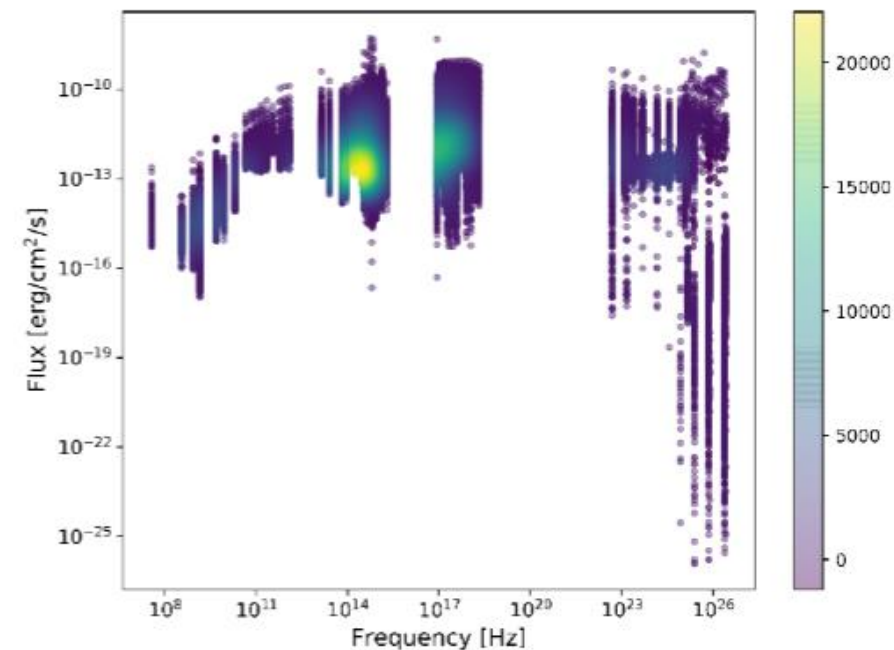
A blazar is an active galactic nucleus (AGN) with a relativistic jet directed very nearly towards an observer. Its standard identification is a manual procedure, usually depends upon heterogeneous multiwavelength coverage.



Artist's impression



(a) Non-Blazars.

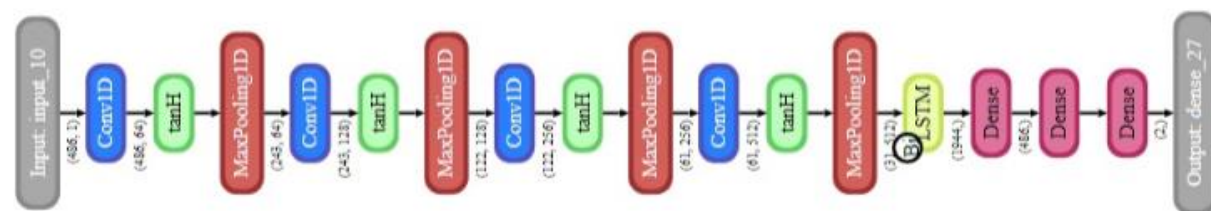
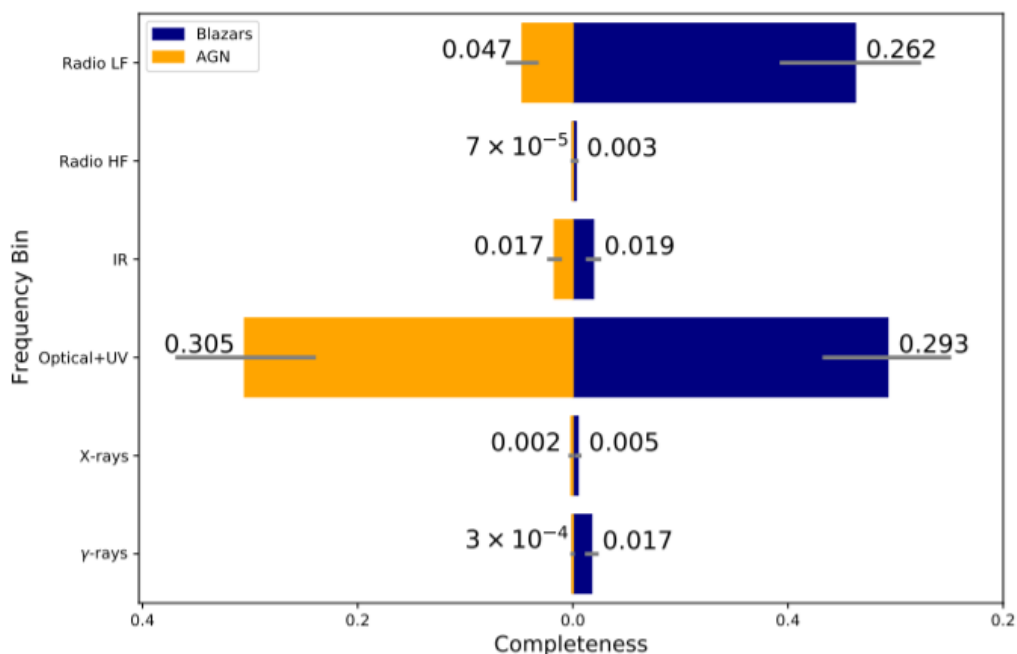


(b) Blazars

[Fraga et al. 2021](#)
[arXiv:2012.15340](#)

RNN for Blazars SED classification

A blazar is an active galactic nucleus (AGN) with a relativistic jet directed very nearly towards an observer. Its standard identification is a manual procedure, usually depends upon heterogeneous multiwavelength coverage. Its highly inhomogeneous sample



[Fraga et al. 2021](#)
[arXiv:2012.15340](#)

Can we make a reliable model of Strong lensing systems with Deep Neural Networks?

We simulate the Lens population from DES and add constraints to derive if they would be observable, e.g. SNR >20. Images resolved.

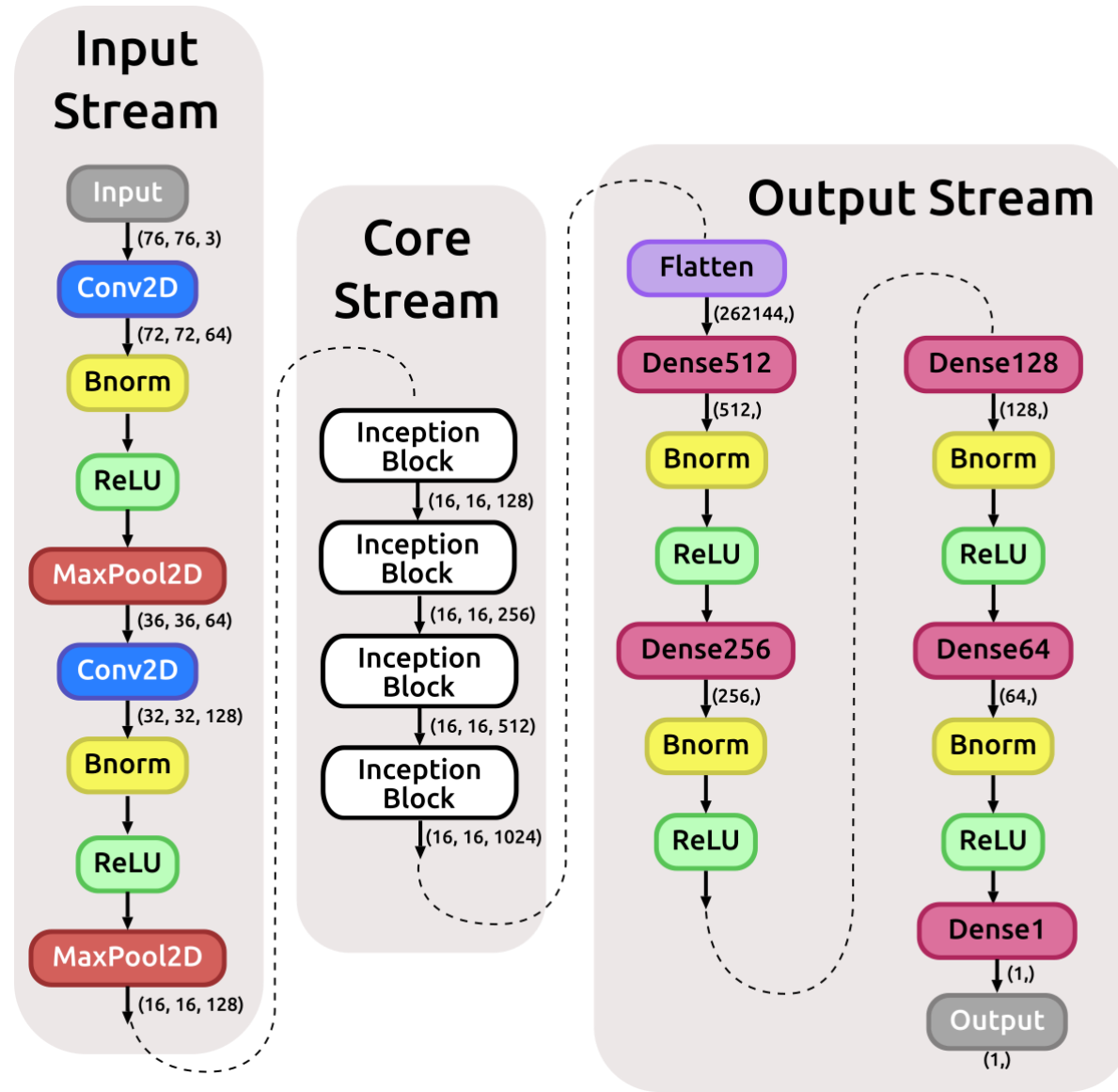
We ended up with 18600 galaxy scale lens systems.

The Lens model is a Singular Isothermal Ellipsoids (SIEs)

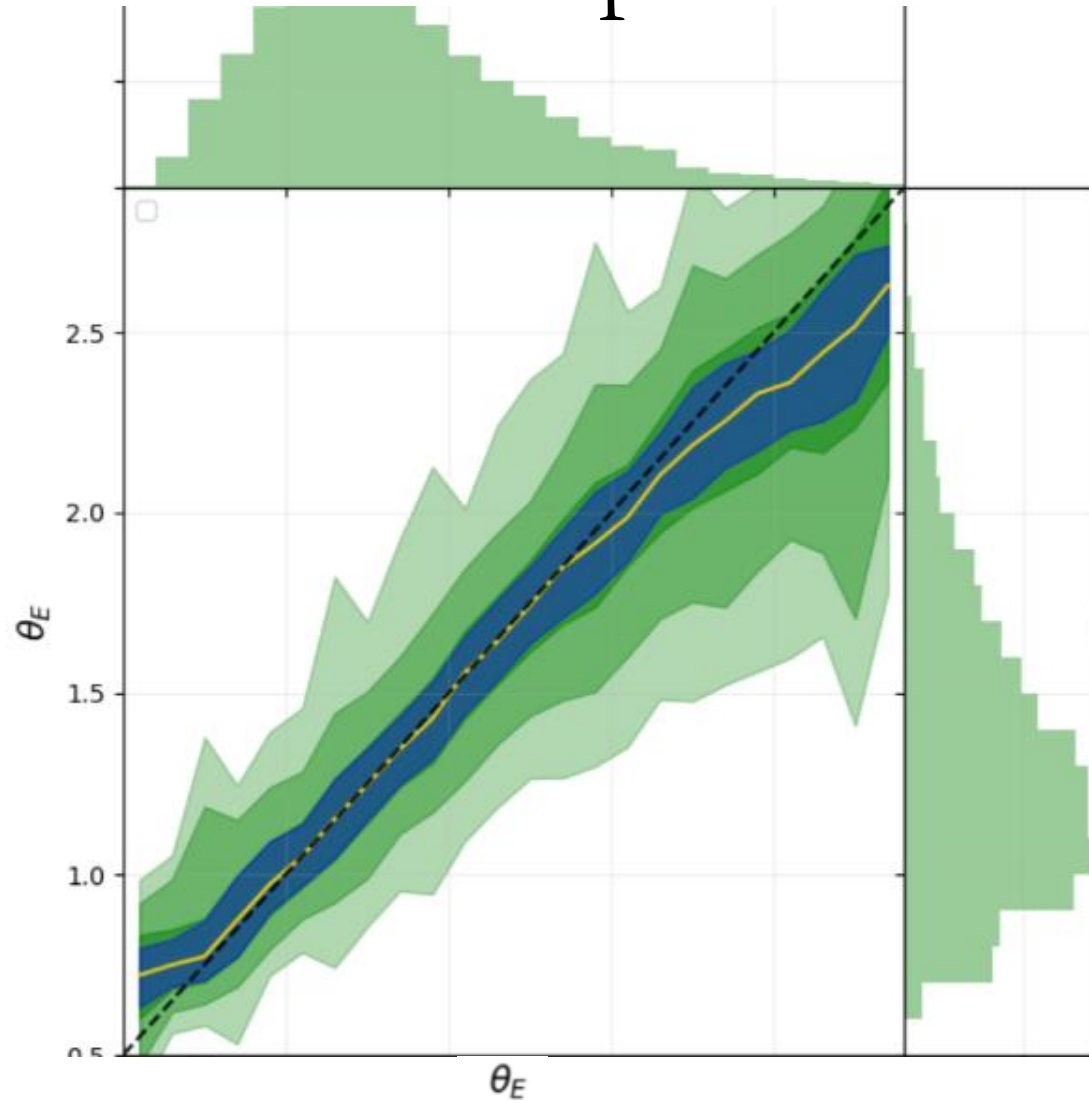
$$\rho(\tilde{r}) = \frac{\sigma_V^2}{2\pi G \tilde{r}^2}.$$

$$\theta_E^{\text{SIS}} = 4\pi \frac{\sigma_V^2}{c^2} \frac{D_{\text{ls}}}{D_s}.$$

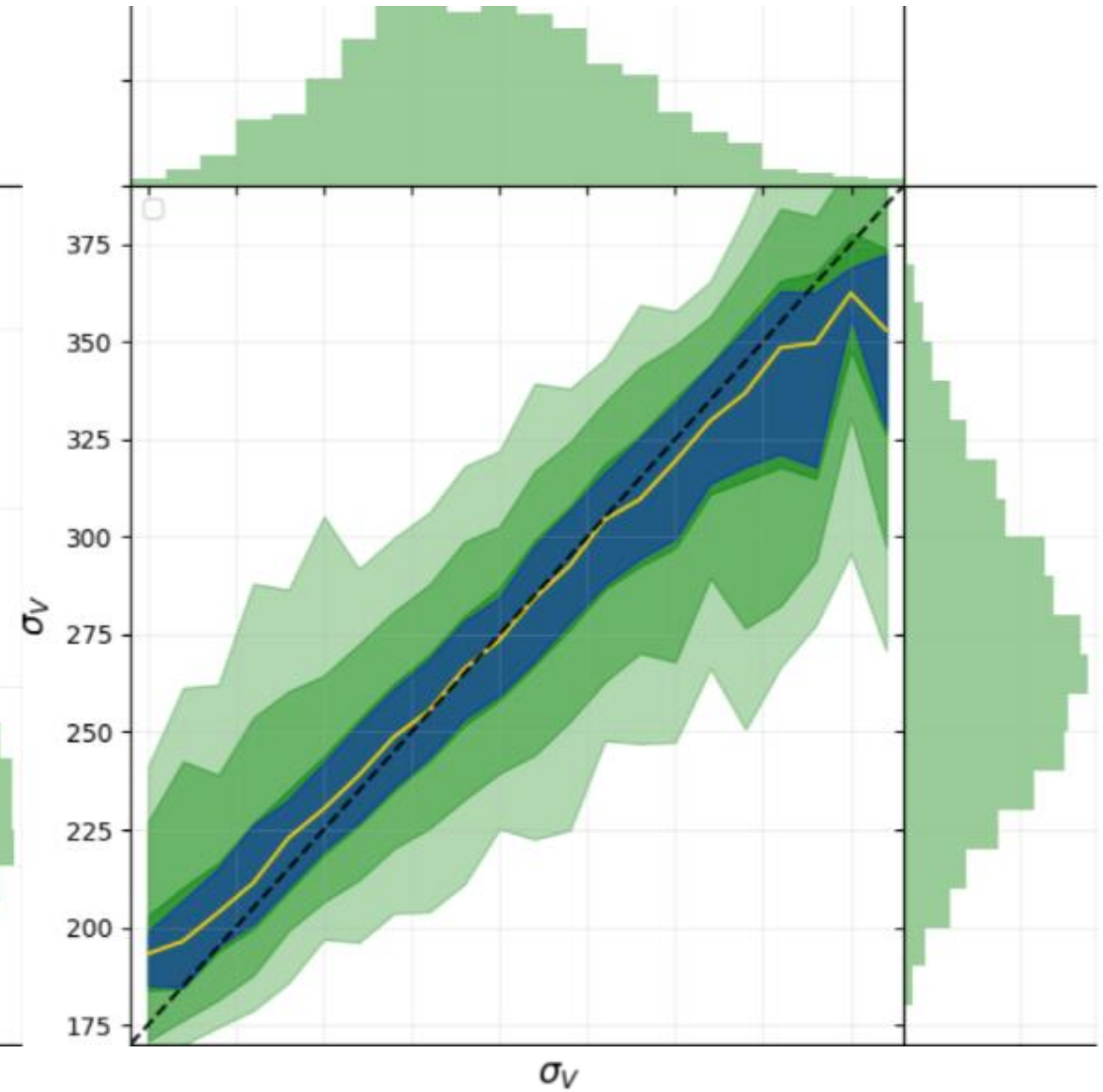
The Deep Learning Model: Inception



One to one line plot



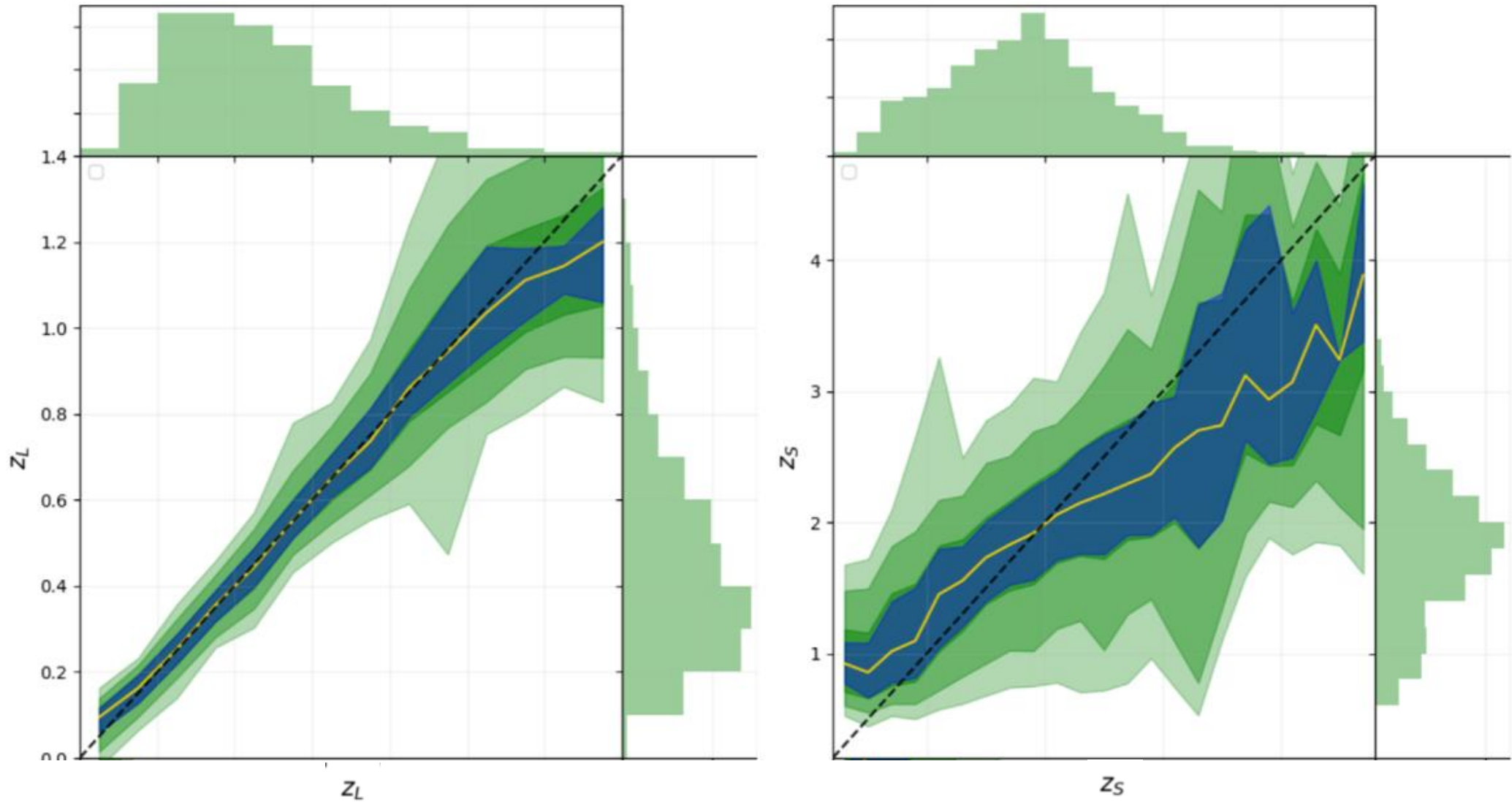
Intuitively correlated with
curvature radius



Connected with mass distribution

$$\rho(\tilde{r}) = \frac{\sigma_V^2}{2\pi G \tilde{r}^2}.$$

One to one line plot



Connected with the distance through cosmological model

Einstein Rings (idealized case)

Singular Isothermal Sphere:

$$\rho(r) = \frac{\sigma_v^2}{2\pi G r^2}$$

slip parameter

$$\gamma = \frac{\Phi}{\Psi}$$

Einstein Ring in modified gravity

$$\theta_E = 4\pi\sigma_{\text{obs}}^2 \left(\frac{1+\gamma}{2} \right) \frac{D_{LS}}{D_S}$$

If $\gamma = 1$, GR is correct. In our simulated sample we assume GR

Measure velocity dispersion + Einstein Radius → Test of Einstein General Relativity

Current results need detailed modelling of the lens + z_s , which is hard to obtain

Results

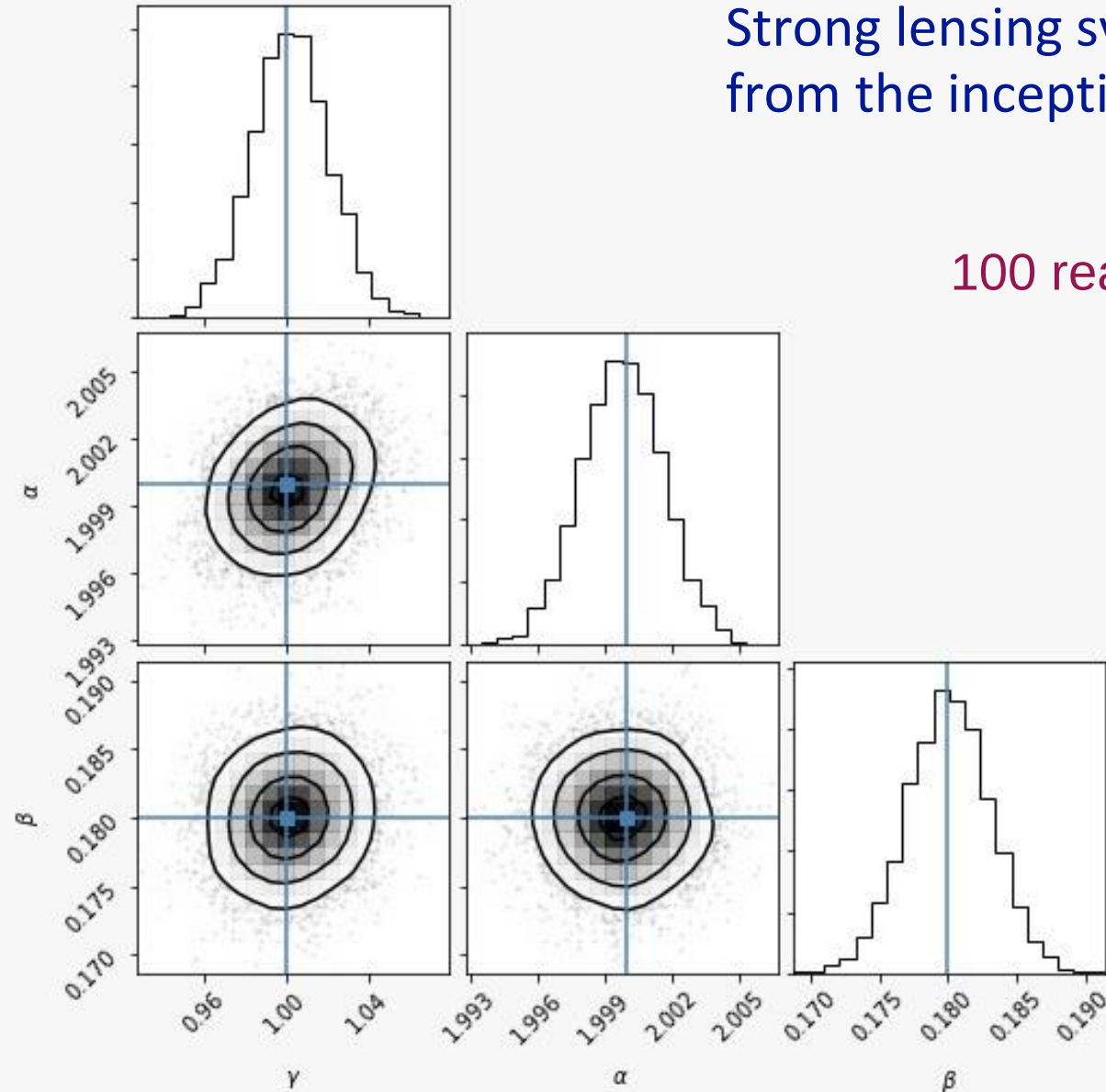
Crisnejo, MM, CB

Strong lensing systems with parameters recovered from the inception neural network

100 realizations of the simulated sample

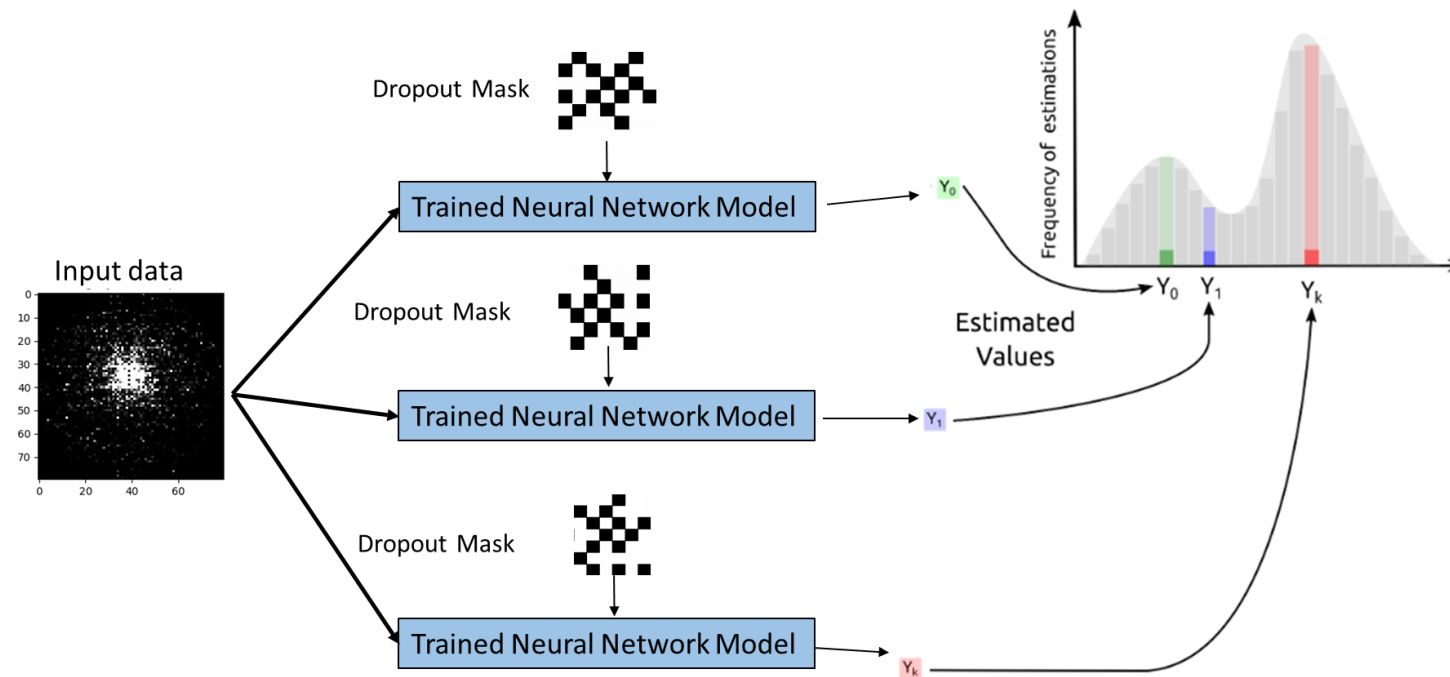
Mean slip parameter

$$\gamma = 1.003 \pm 0.019$$



Error estimative – Monte Carlo Dropout

Neural Nets are Universal approximators. Thus, a small perturbation in the network weights are considered a sampling over the space of functions.

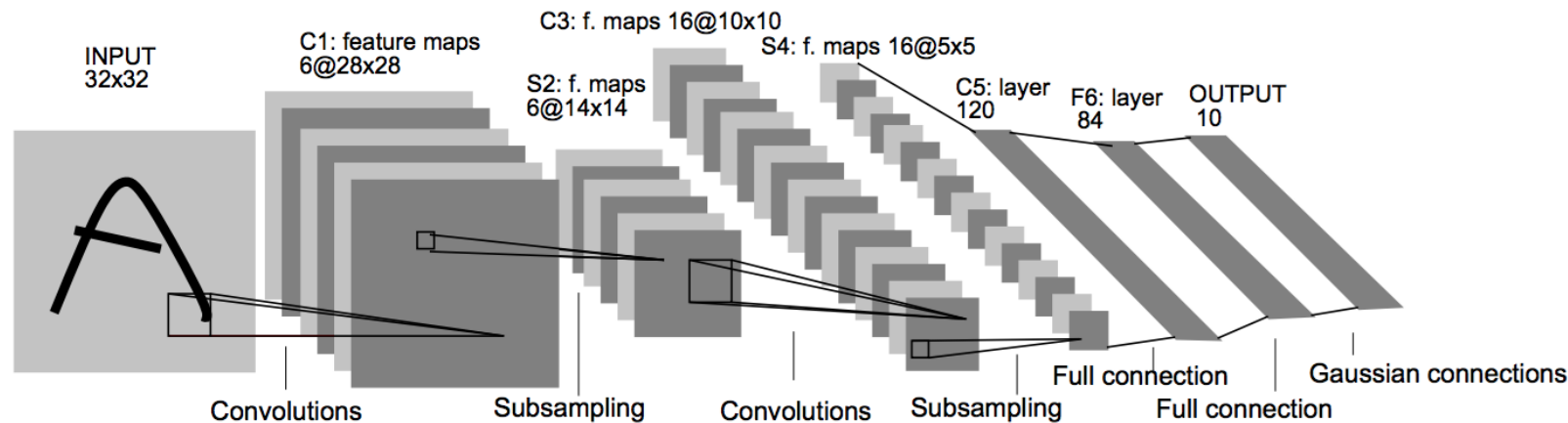


Therefore, we may define a PDF for each input. This PDF encapsulates the systematic uncertainty due to the Deep Learning Model, known as Epistemic Uncertainty. We also make a regression to predict the level of statistical uncertainty.

A little bit of historical Nets ...

LeNet-5

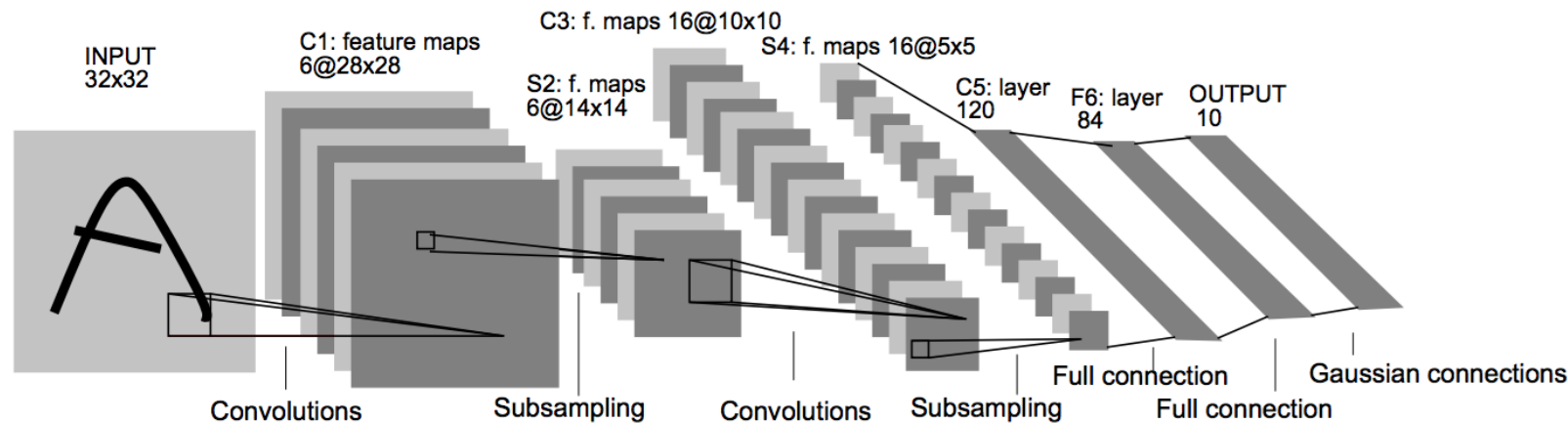
Yann Lecun's LeNet-5 model was developed in 1998 to identify handwritten digits for zip code recognition in the postal service. This pioneering model largely introduced the convolutional neural network as we know it today.



A little bit of historical Nets ...

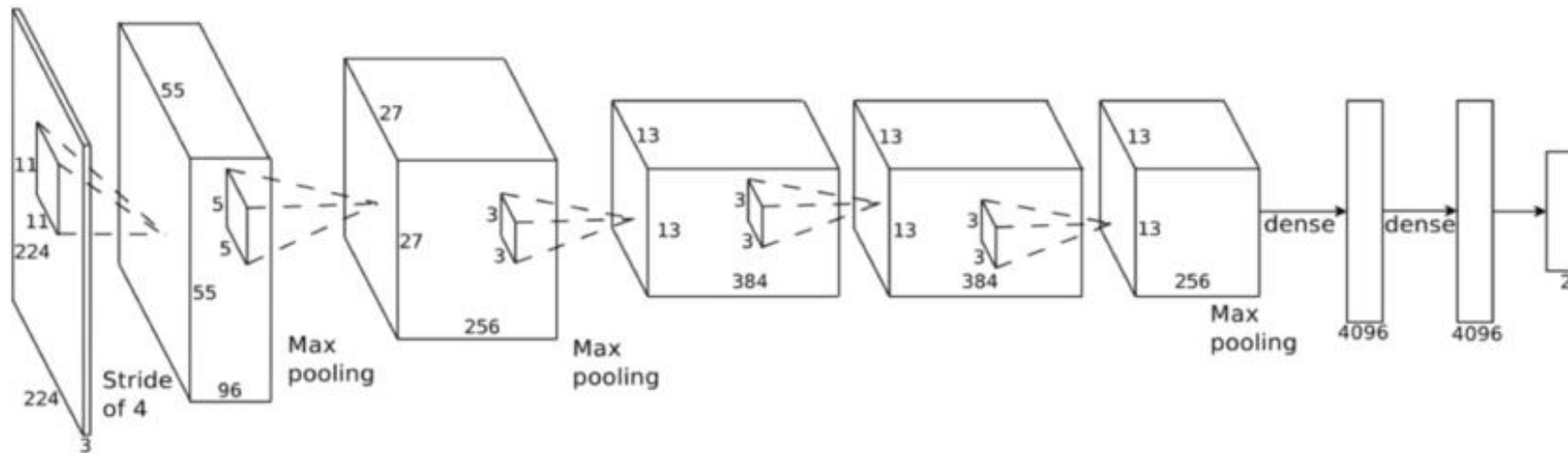
The subsampling layers use a form of average pooling.

Parameters: 60,000



A little bit of historical Nets ...

AlexNet was developed by Alex Krizhevsky et al. in 2012 to compete in the ImageNet competition. The general architecture is quite similar to LeNet-5, although this model is considerably larger. The success of this model (which took first place in the 2012 ImageNet competition) convinced a lot of the computer vision community to take a serious look at deep learning for computer vision tasks.

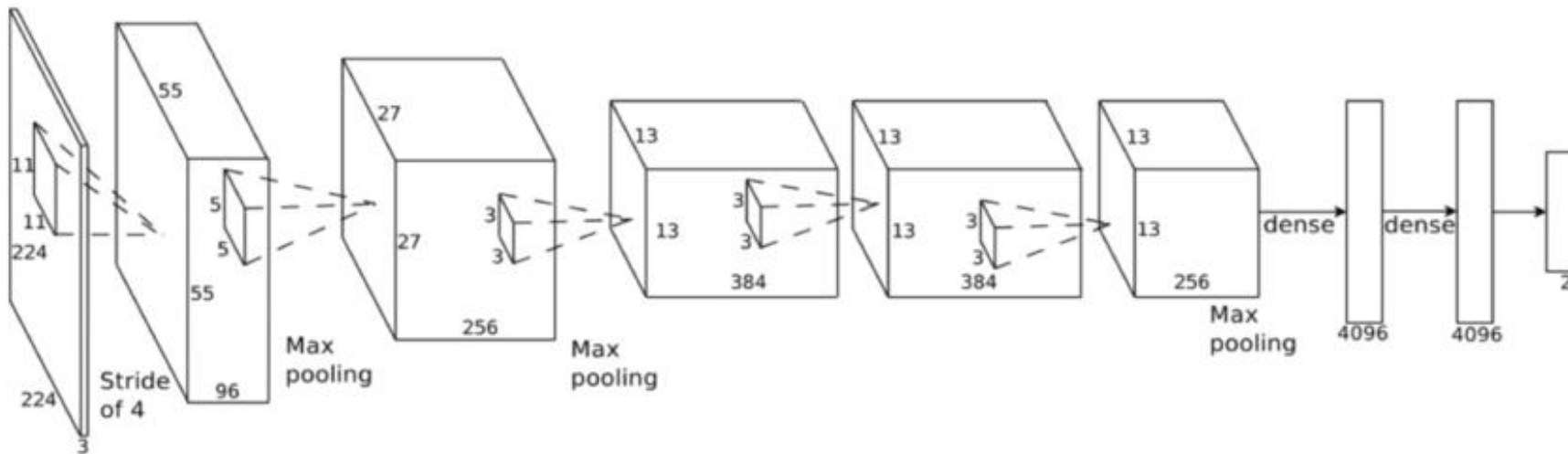


[doi:10.1145/3065386](https://doi.org/10.1145/3065386)

A little bit of historical Nets ...

AlexNet → 60 Million parameters!!!!

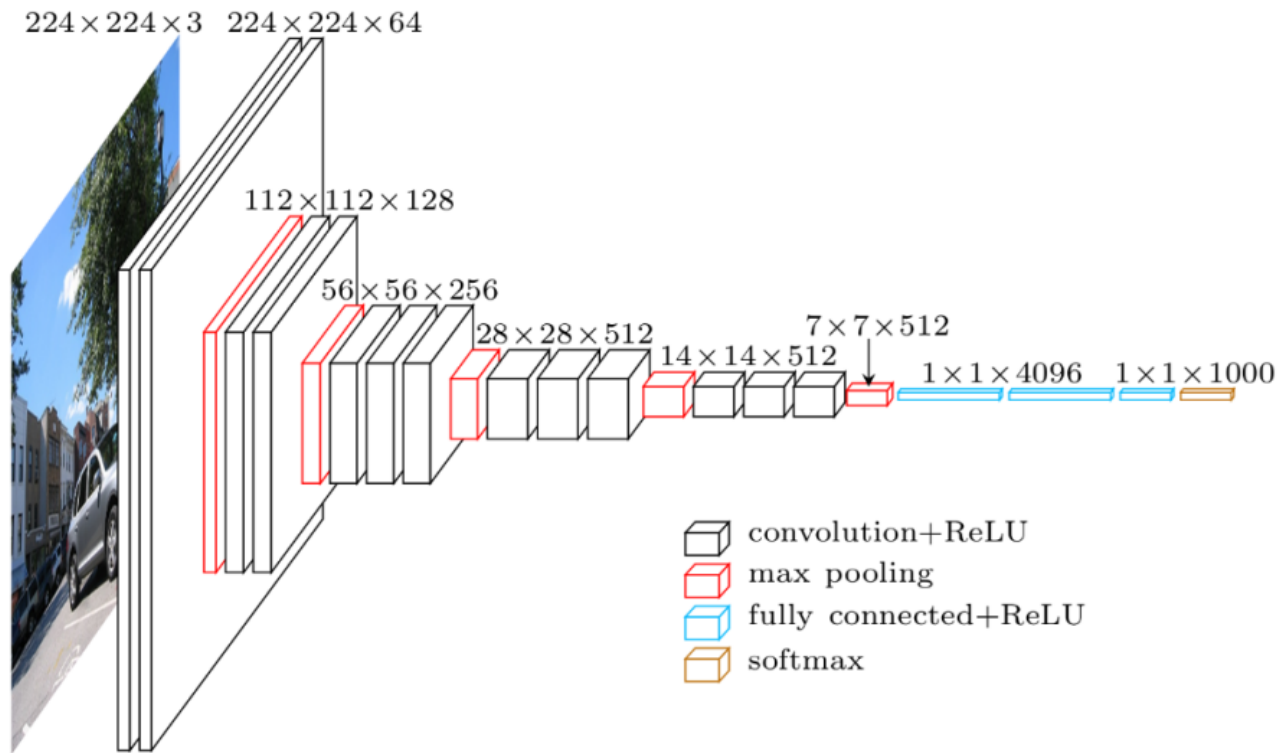
650,000 neurons, consists of five convolutional layers, some of which are followed by max-pooling layers, and three globally-connected layers with a final 1000-way softmax. It was trained on two NVIDIA GPUs for about a week.



[doi:10.1145/3065386](https://doi.org/10.1145/3065386)

A little bit of historical Nets ...

VGG-16 (2014)→ 138 Million parameters!!!!



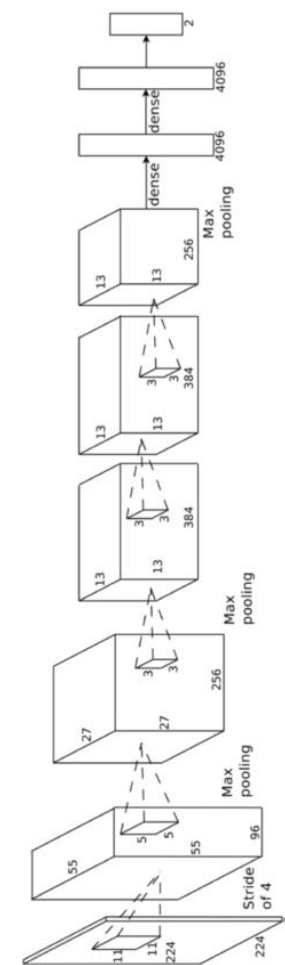
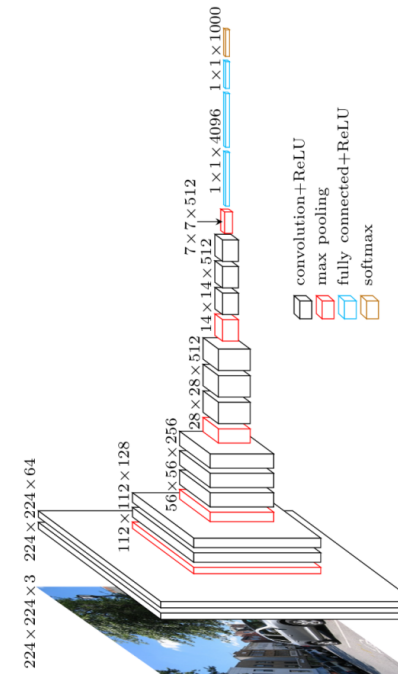
<https://arxiv.org/abs/1409.1556>

What can we do with such BFT (Best-Fitted and Trained?) models today?

Clustering!!!! They do know how to extract features.
Unsupervised learning
Transfer Learning



Ben Kenobi - Old and Wise
lots of optimized parameters



Going Deeper with Convolutions - AKA - Inception paper

Bigger size typically means a larger number of parameters, which makes the enlarged network more prone to overfitting, especially if the number of labeled examples in the training set is limited. (...) The other drawback of uniformly increased network size is the dramatically increased use of computational resources. For example, in a deep vision network, if two convolutional layers are chained, any uniform increase in the number of their filters results in a **quadratic increase of computation**. If the added capacity is used inefficiently (for example, if most weights end up to be close to zero), then much of the computation is wasted.(...).



Going Deeper with Convolutions - AKA - Inception paper

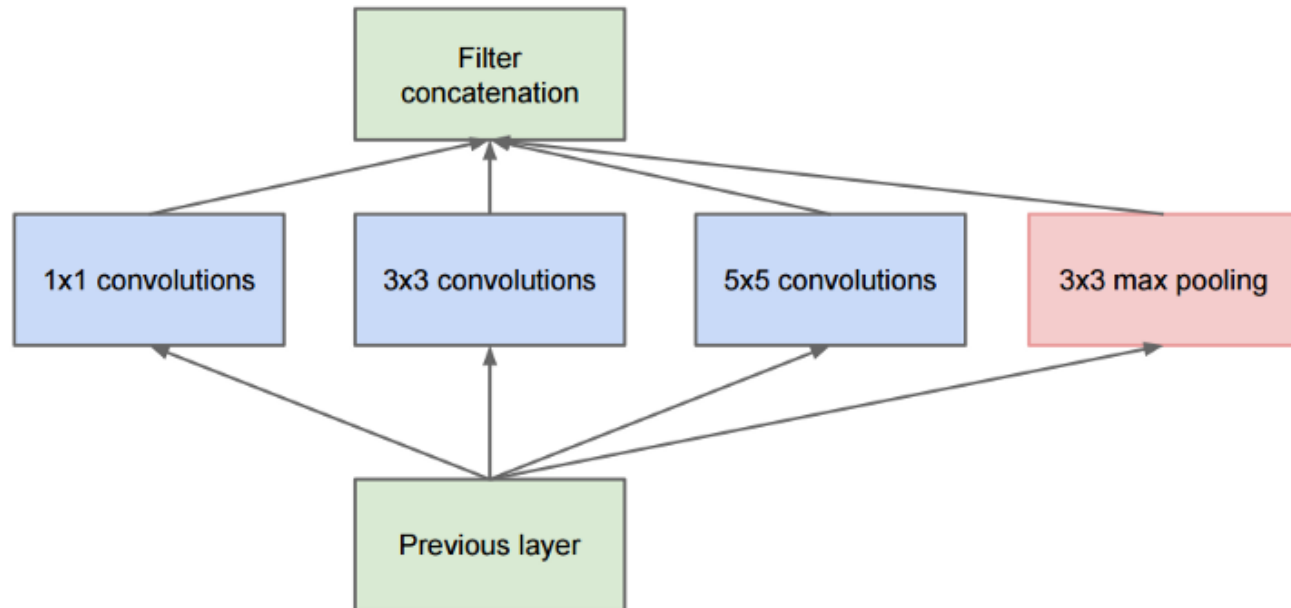
A fundamental way of solving both of these issues would be **to introduce sparsity and replace the fully connected layers by the sparse ones, even inside the convolutions.** Besides mimicking biological systems, this would also have the advantage of firmer theoretical underpinnings due to the groundbreaking work of Arora et al. [2]. **Their main result states that if the probability distribution of the dataset is representable by a large, very sparse deep neural network, then the optimal network topology can be constructed layer after layer by analyzing the correlation statistics of the preceding layer activations and clustering neurons with highly correlated outputs.**



The Main (Inception) Idea...

GoogLeNet (2014)

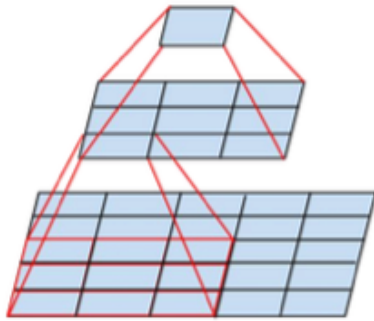
The idea is that you don't need to know in advance if it was better to do, for example, a 3×3 then a 5×5 . Instead, just do all the convolutions and let the model pick what's best. Additionally, this architecture allows the model to recover both local feature via smaller convolutions and high abstracted features with larger convolutions.



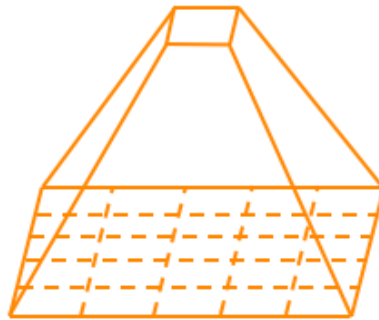
The Main (Inception) Idea...

GoogLeNet (2014)

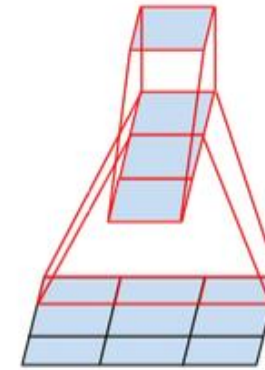
5 million (V1) and 23 million (V3)



two successive
3x3 convolutions



5x5 convolution



3x3 convolutions could be further deconstructed into successive 3x1 and 1x3 convolutions.

Generalizing this insight, we can more efficiently compute an $n \times n$ convolution as a $1 \times n$ convolution followed by a $n \times 1$ convolution..

What to choose?

Merge layers

Add/Merge



Subtract



Multiply



Average



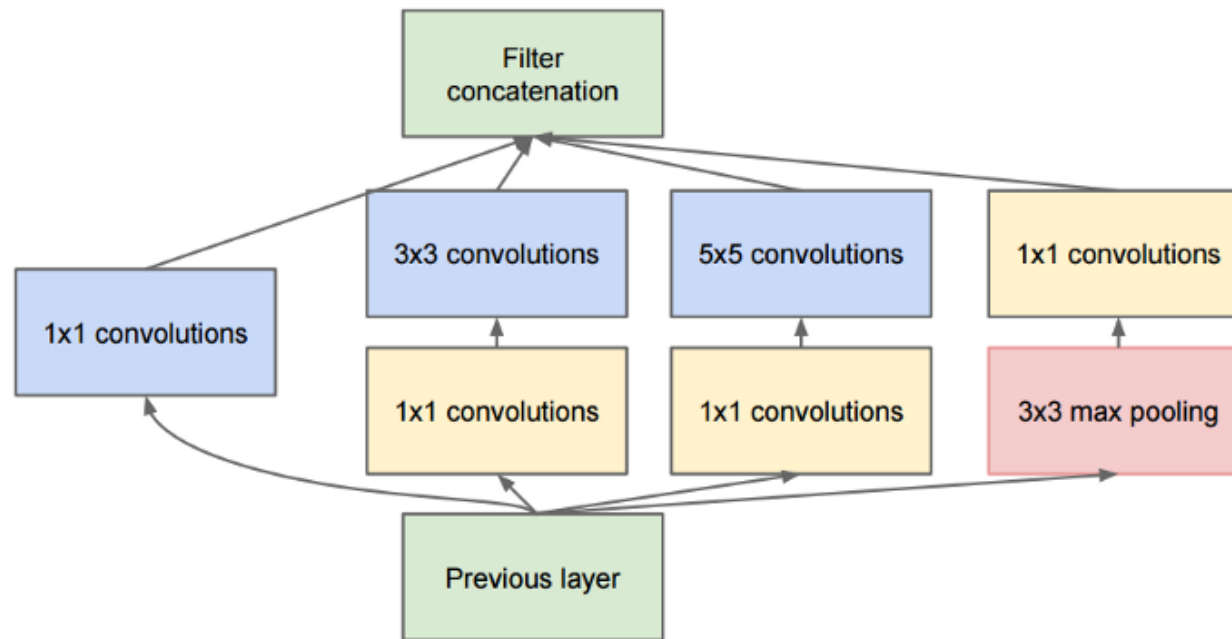
Maximum



Concatenate

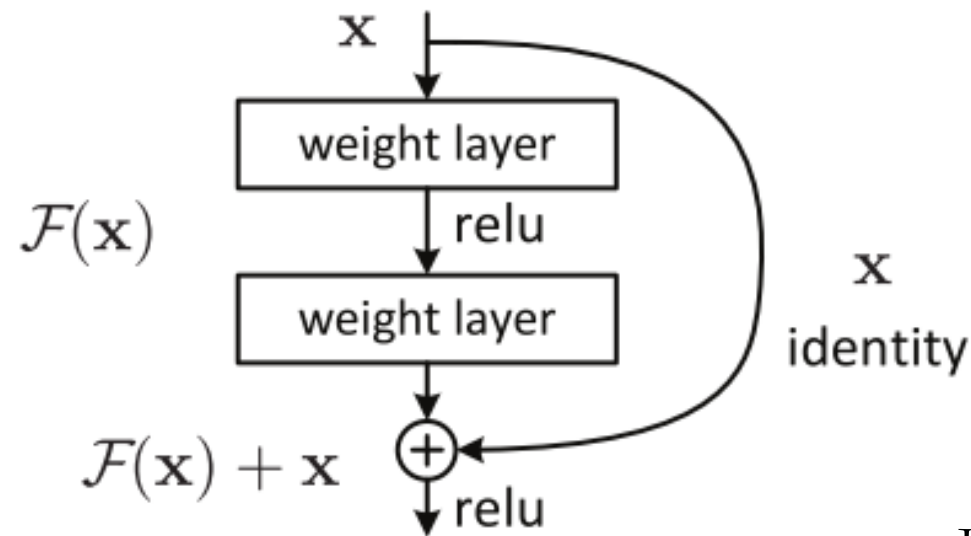


Dot



We still have millions of parameters to fit!!!! We still need some ideas to prevent overfitting

ResNet Block



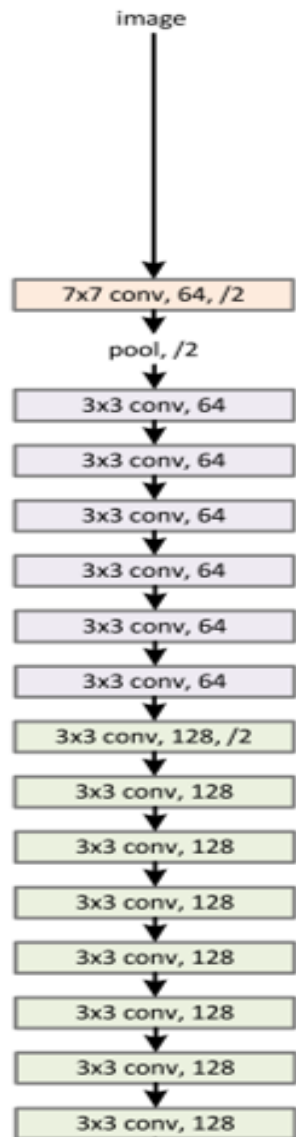
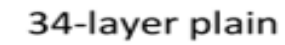
$$H(x) = F(x) + x$$

The author's hypothesis is that it is easy to optimize the residual mapping function $F(x)$ than to optimize the original, unreferenced mapping .

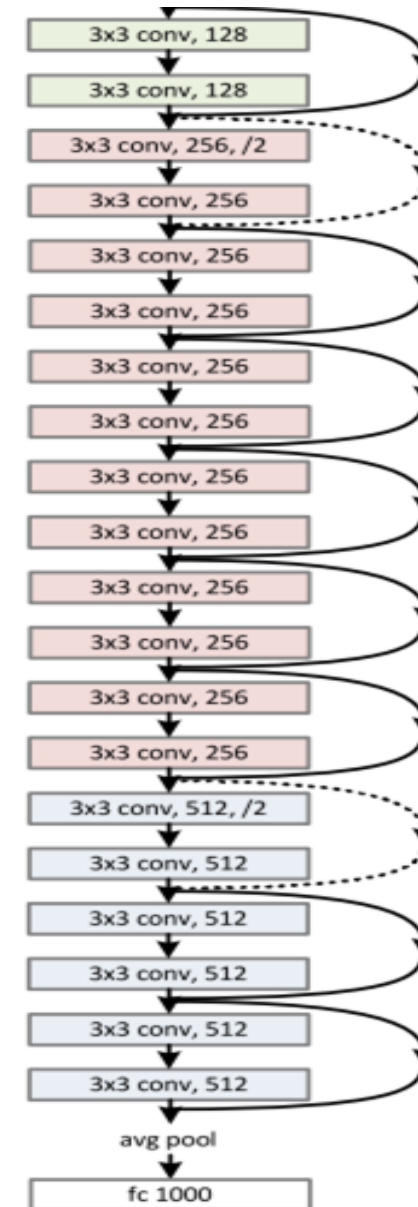
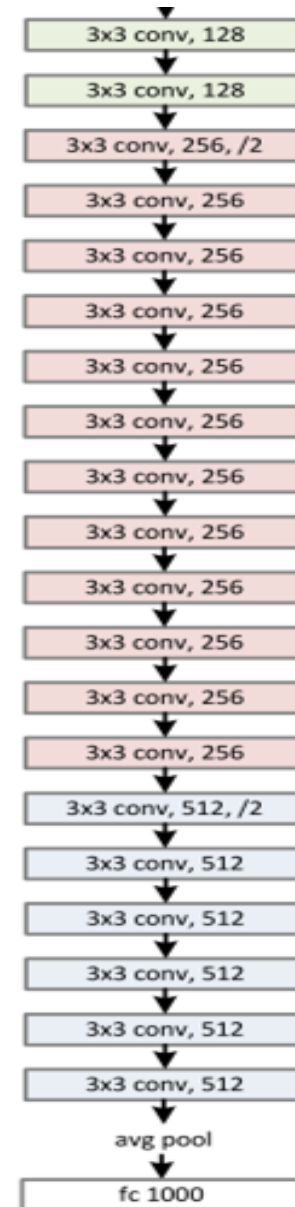
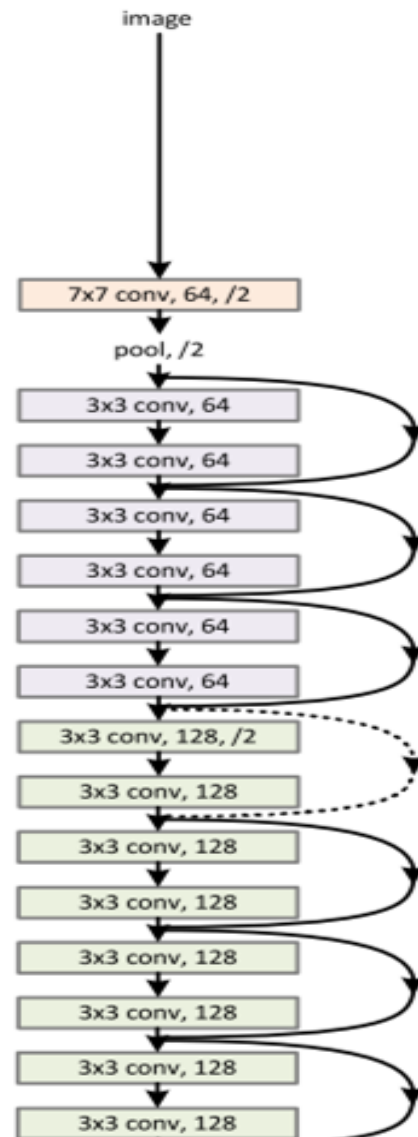
If the identity mapping is optimal, We can easily push the residuals to zero ($F(x) = 0$) than to fit an identity mapping (x , input=output) by a stack of non-linear layers.

It also put a new light on the vanishing gradient problem...

Residual Neural Networks



34-layer residual



Residual Neural Networks

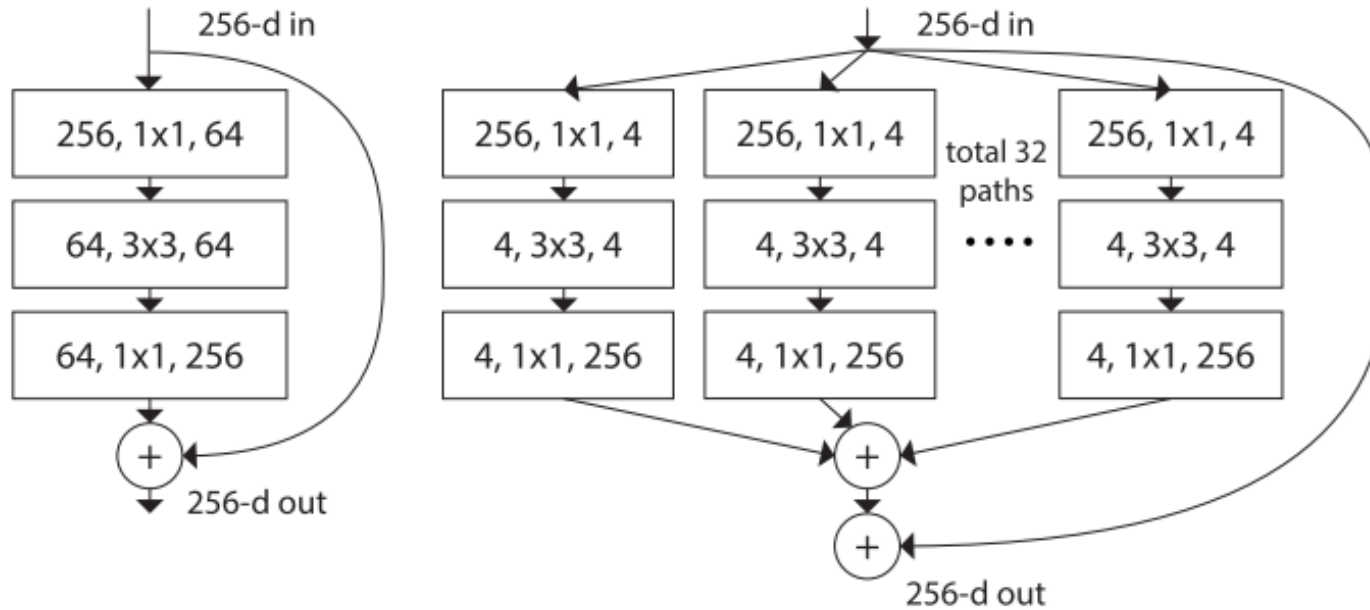
- Won 1st place in the ILSVRC 2015 classification competition with top-5 error rate of 3.57% (An ensemble model)
- Won the 1st place in ILSVRC and COCO 2015 competition in ImageNet Detection, ImageNet localization, Coco detection and Coco segmentation.
- Replacing VGG-16 layers in Faster R-CNN with ResNet-101. They observed a relative improvements of 28%
- Efficiently trained networks with 100 layers and 1000 layers also.
- ResNet Network Converges faster compared to plain counterpart of it.
- Identity vs Projection shortcuts. Very small incremental gains using projection shortcuts in all the layers. So all ResNet blocks use only Identity shortcuts with Projections shortcuts used only when the dimensions changes.

How the other Nets were in ILSVRC 2015

method	top-1 err.	top-5 err.
VGG [41] (ILSVRC'14)	-	8.43 [†]
GoogLeNet [44] (ILSVRC'14)	-	7.89
VGG [41] (v5)	24.4	7.1
PReLU-net [13]	21.59	5.71
BN-inception [16]	21.99	5.81
ResNet-34 B	21.84	5.71
ResNet-34 C	21.53	5.60
ResNet-50	20.74	5.25
ResNet-101	19.87	4.60
ResNet-152	19.38	4.49

Table 4. Error rates (%) of **single-model** results on the ImageNet validation set (except [†] reported on the test set).

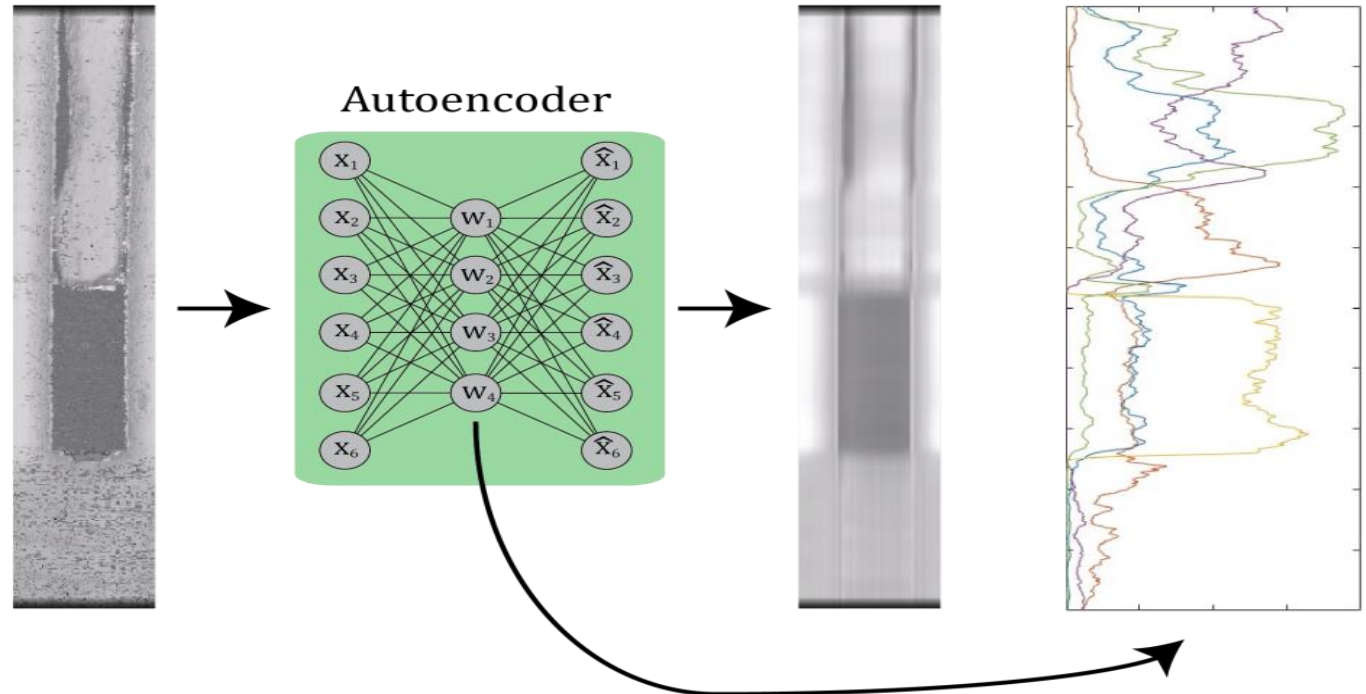
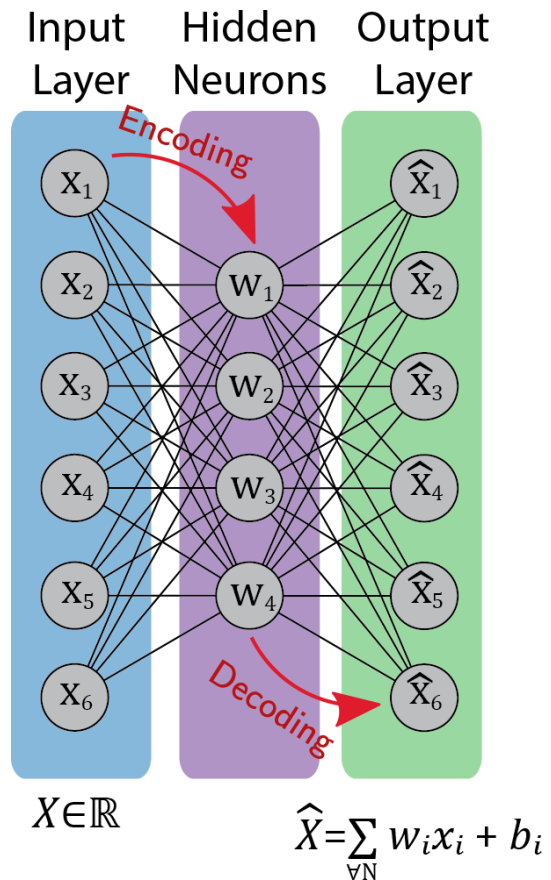
ResNext



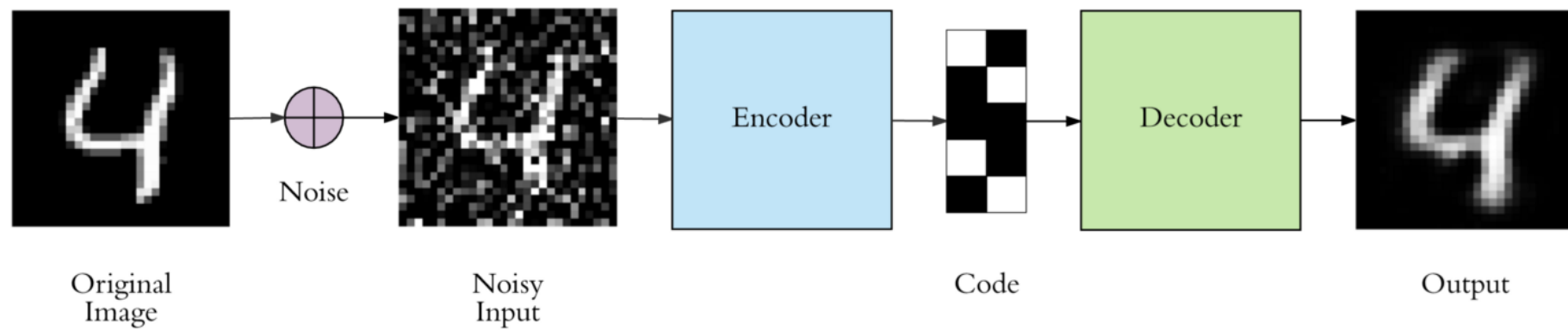
- Similar to Inception. However, the outputs of different paths are merged by adding them together, in the Inception they are concatenated.
- In the Inception paper each path is different (1×1 , 3×3 and 5×5 convolution) and in the ResNext all paths share the same topology.

What is an AutoEncoder?

- ANN aim to reproduce input data.
- **Encode/Decode** strategy.
- **IDENTITY FUNCTION** → Hidden Features

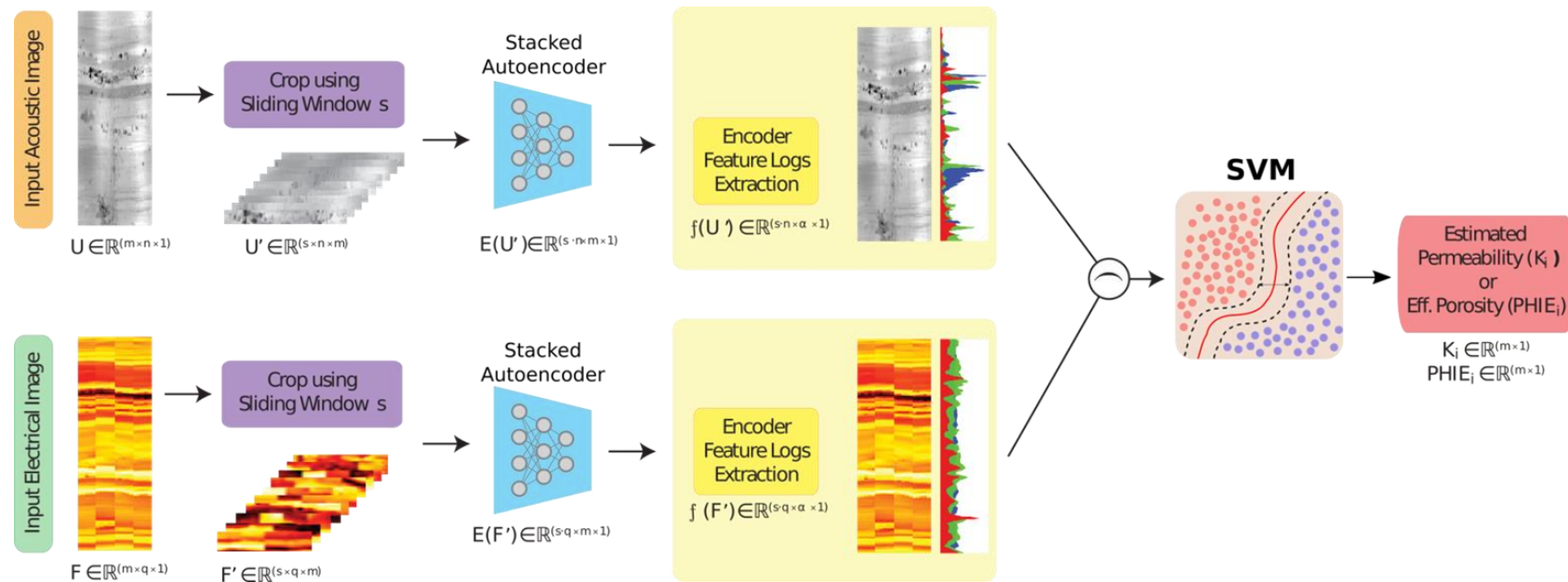


Denoising

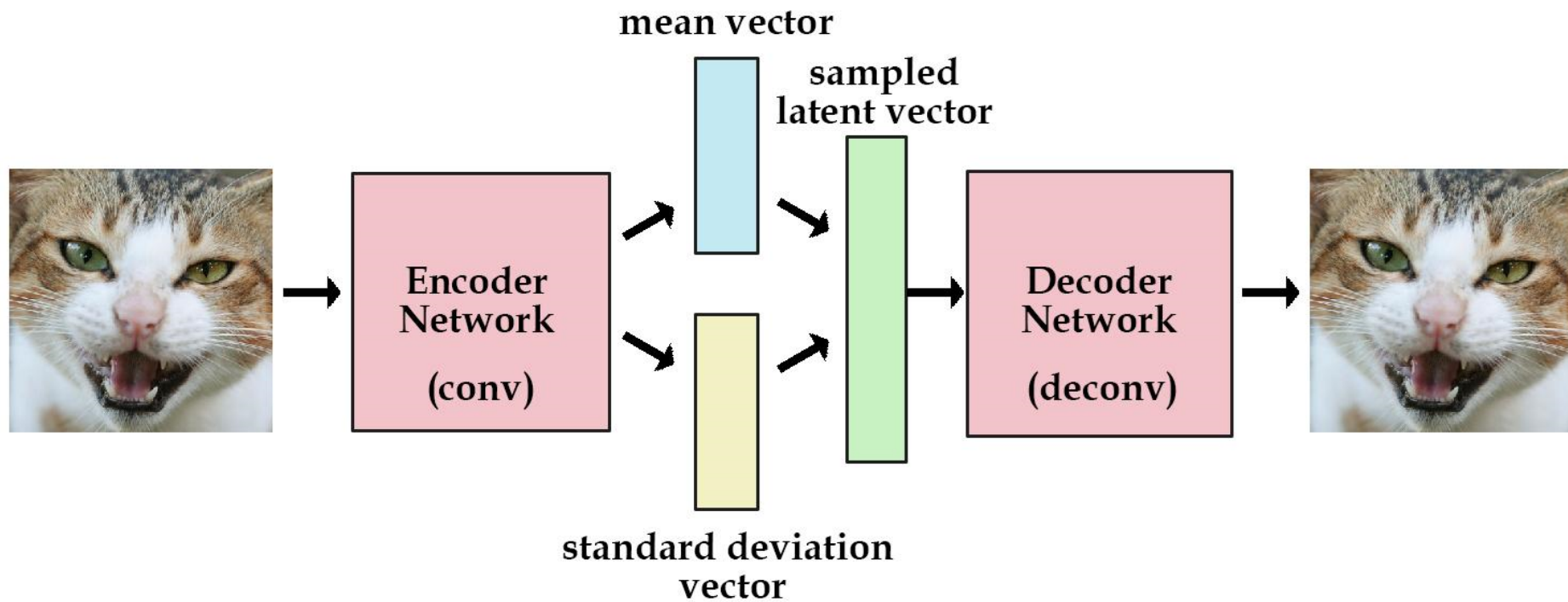


Regression with AE? But Why?

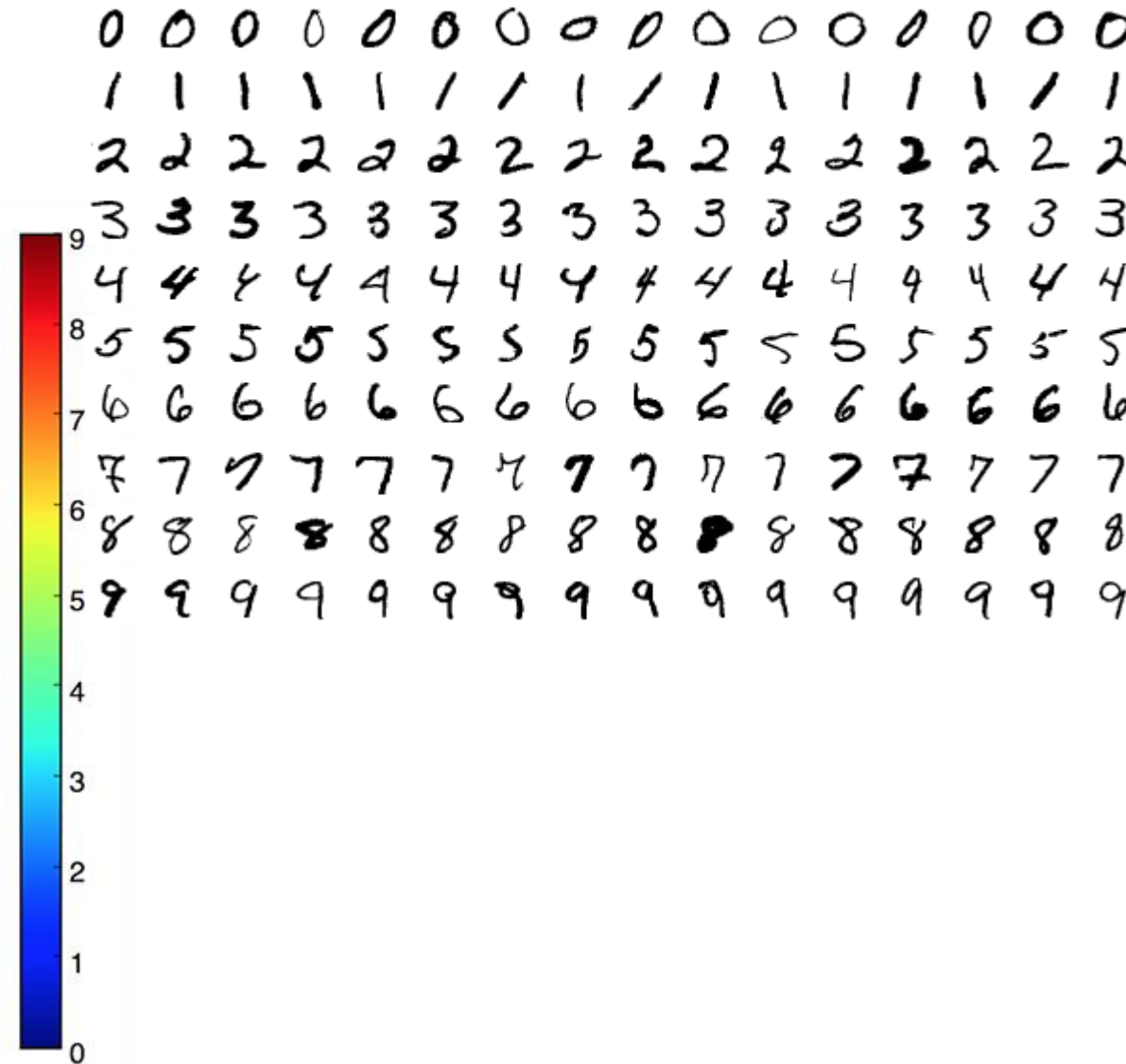
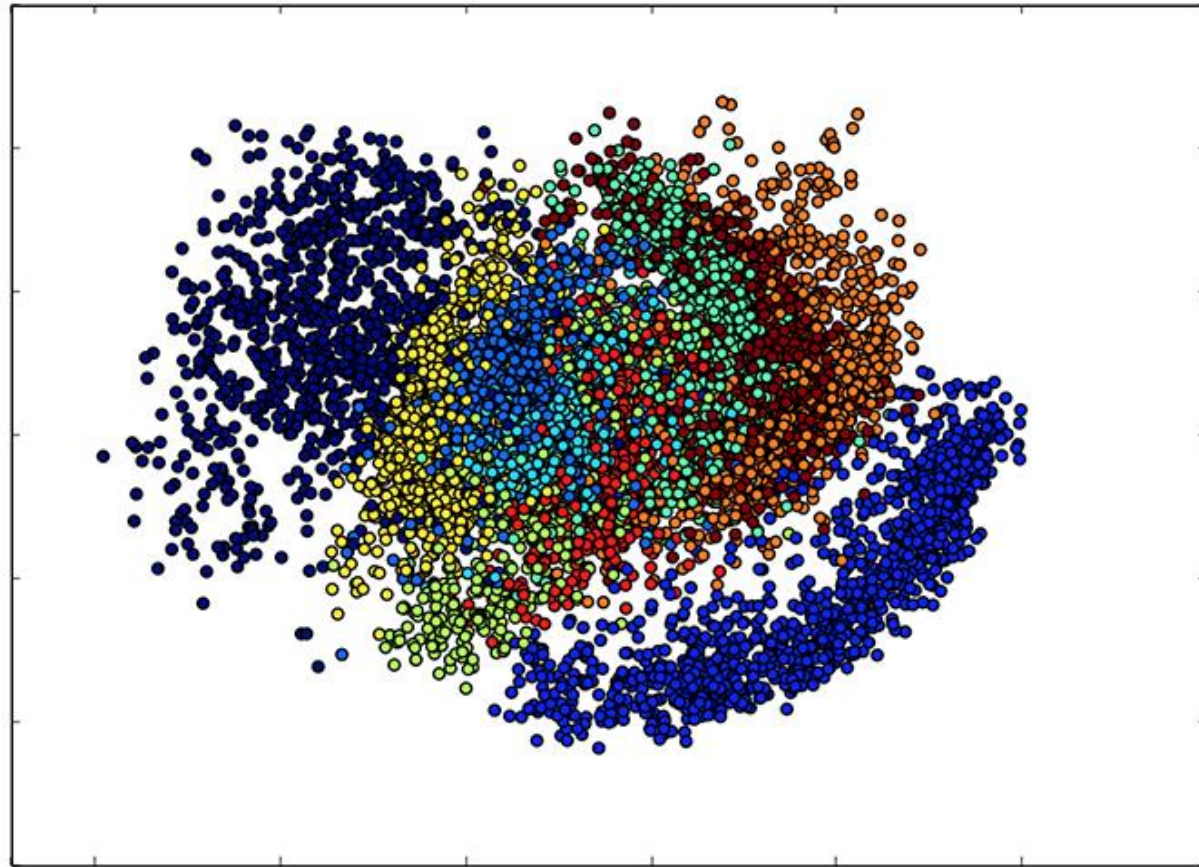
- Small datasets
- “Empty” datasets



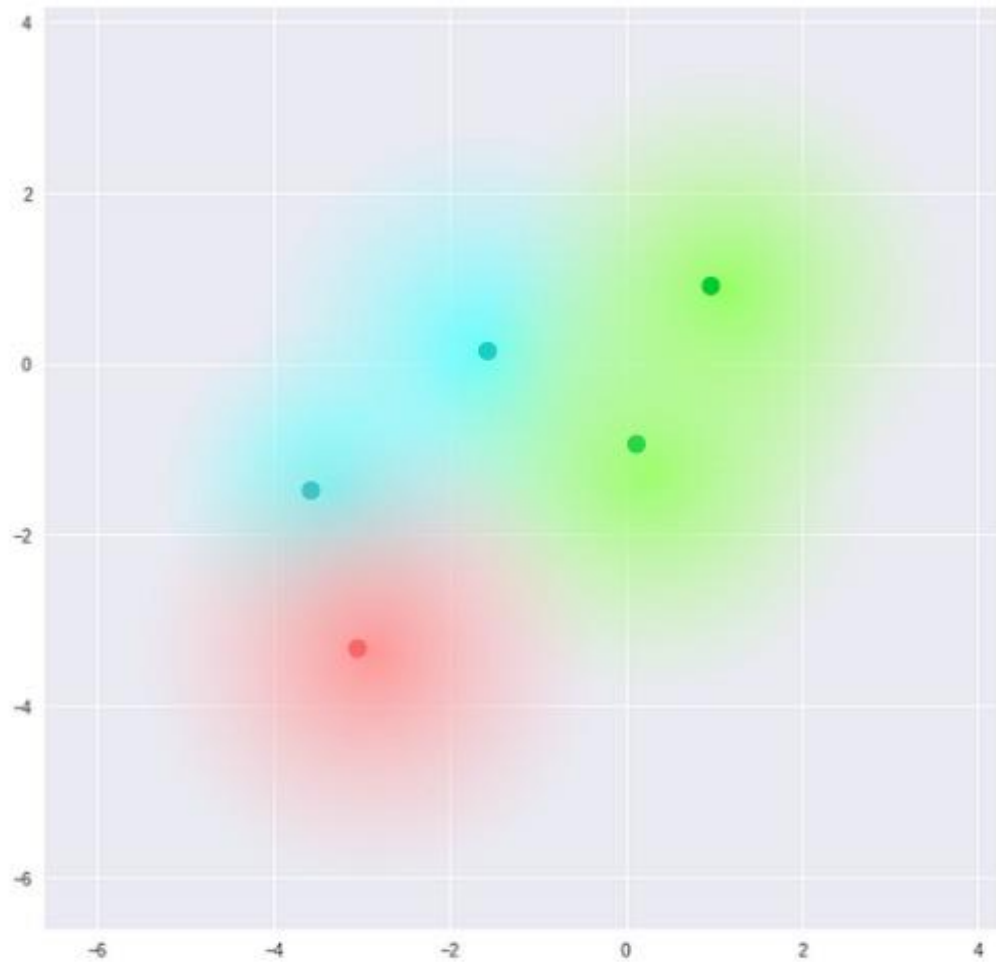
Variational AE



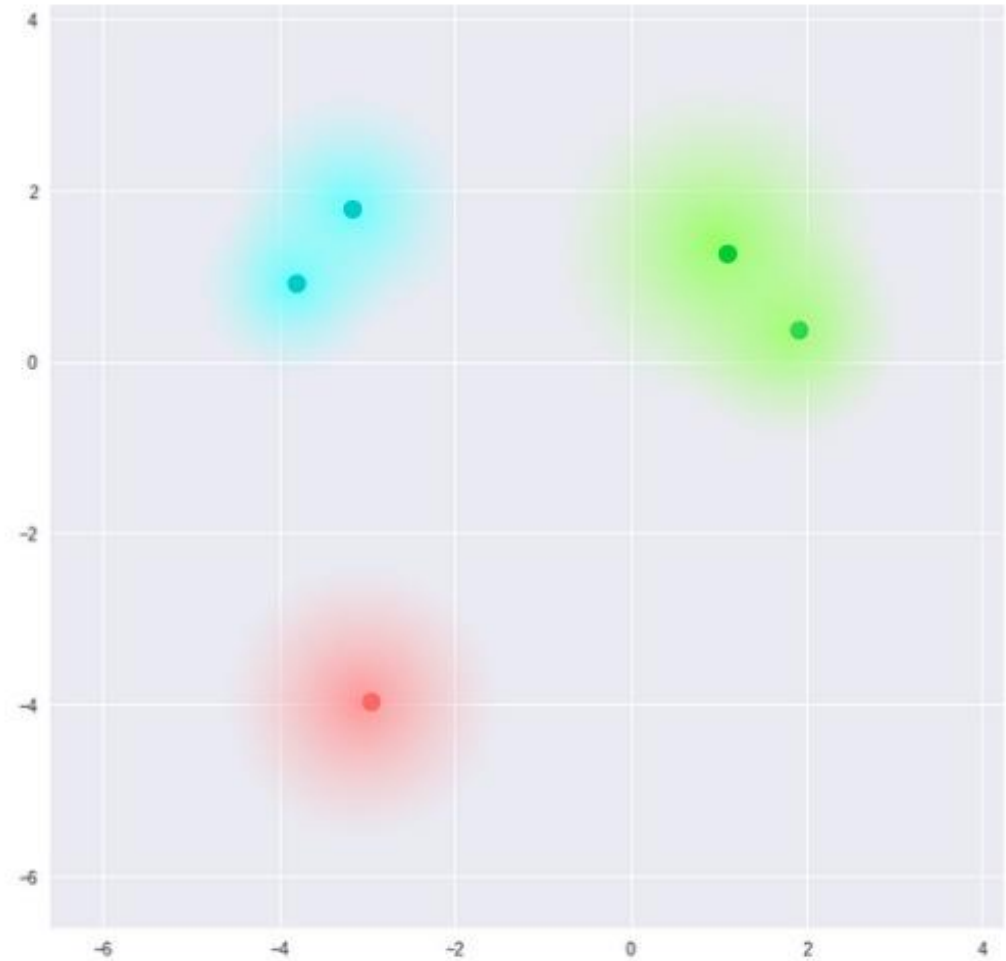
Auto Encoder



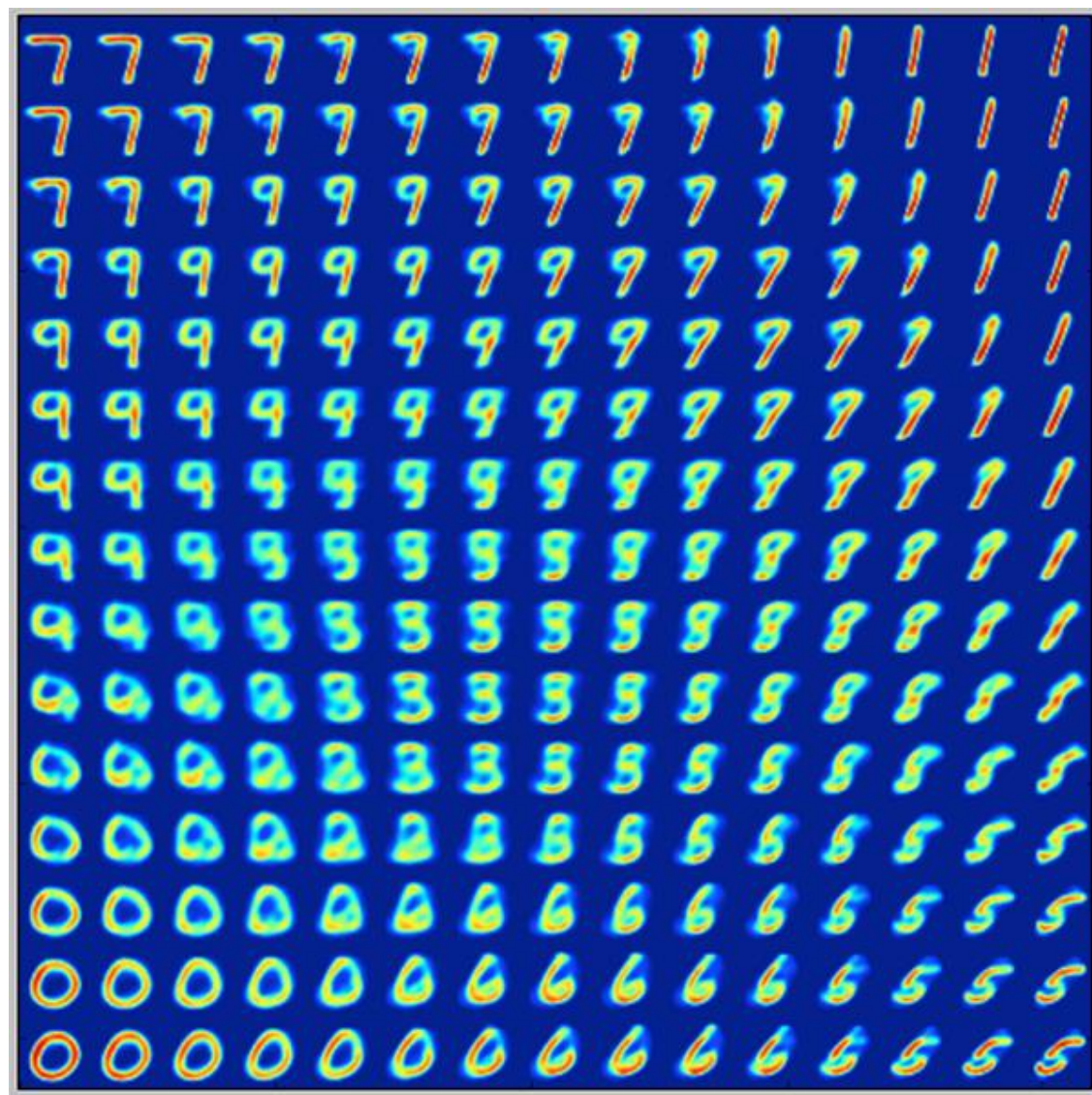
Variational AE – Generative Model



What we require



What we may inadvertently end up with

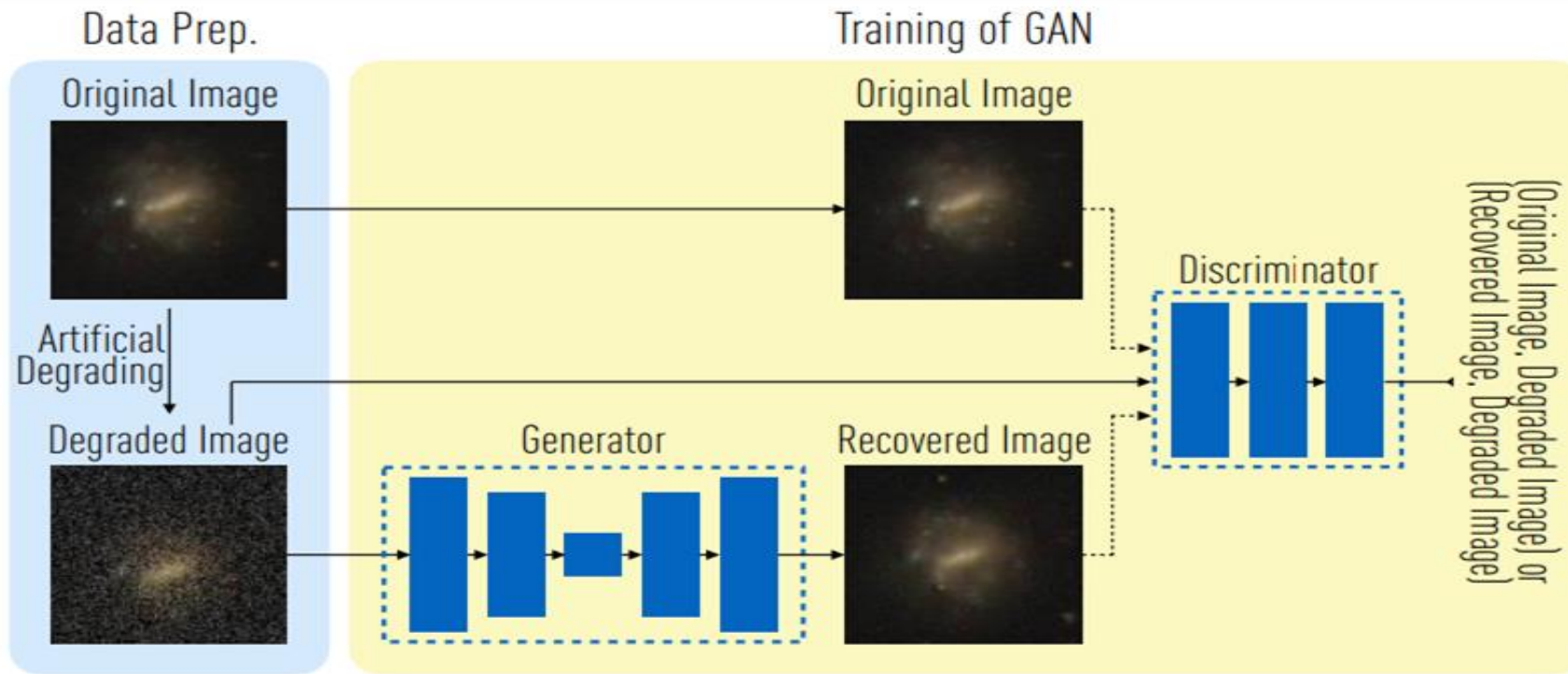


Generative Adversarial Networks

First proposed by Goodfellow et al. 2014 arXiv:1406.2661

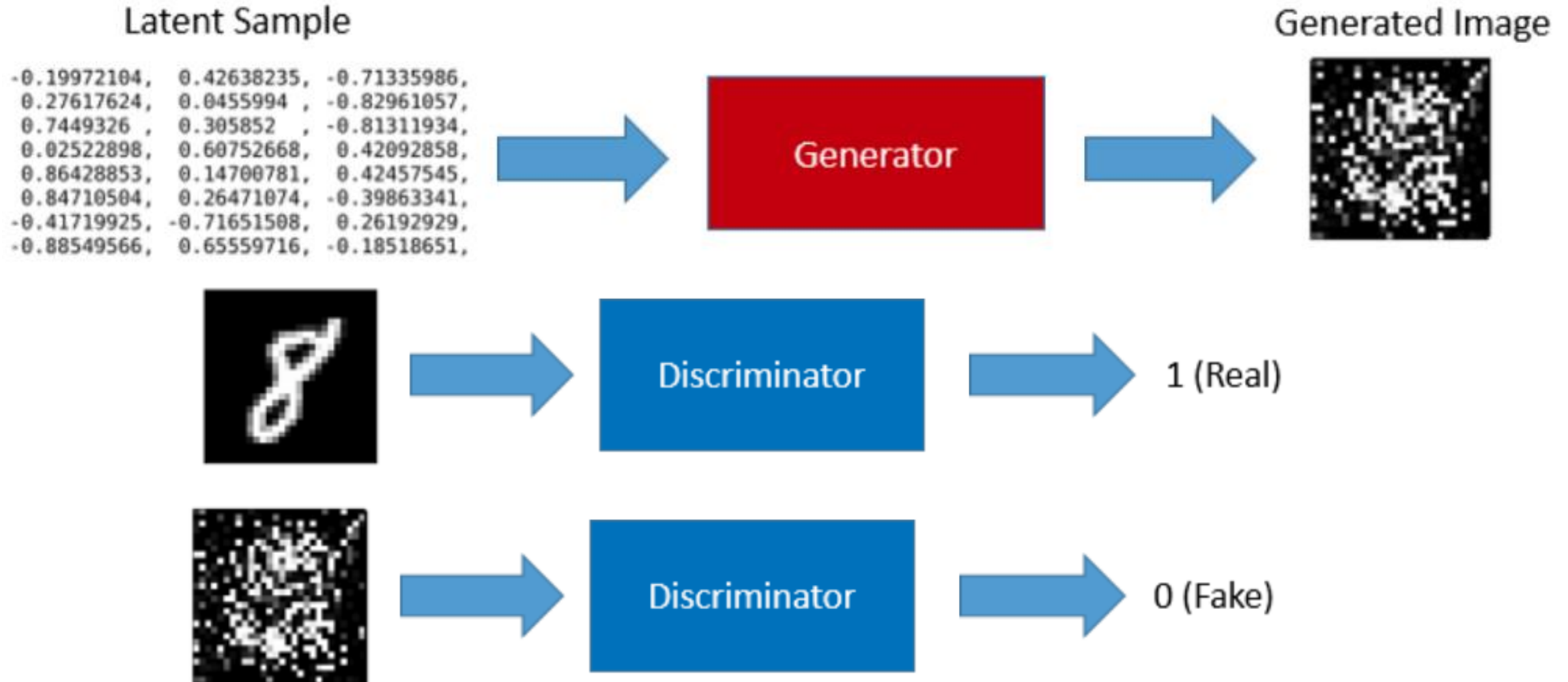
The Main idea is to make two networks to be adversaries.

(arXiv:1702.00403v1).



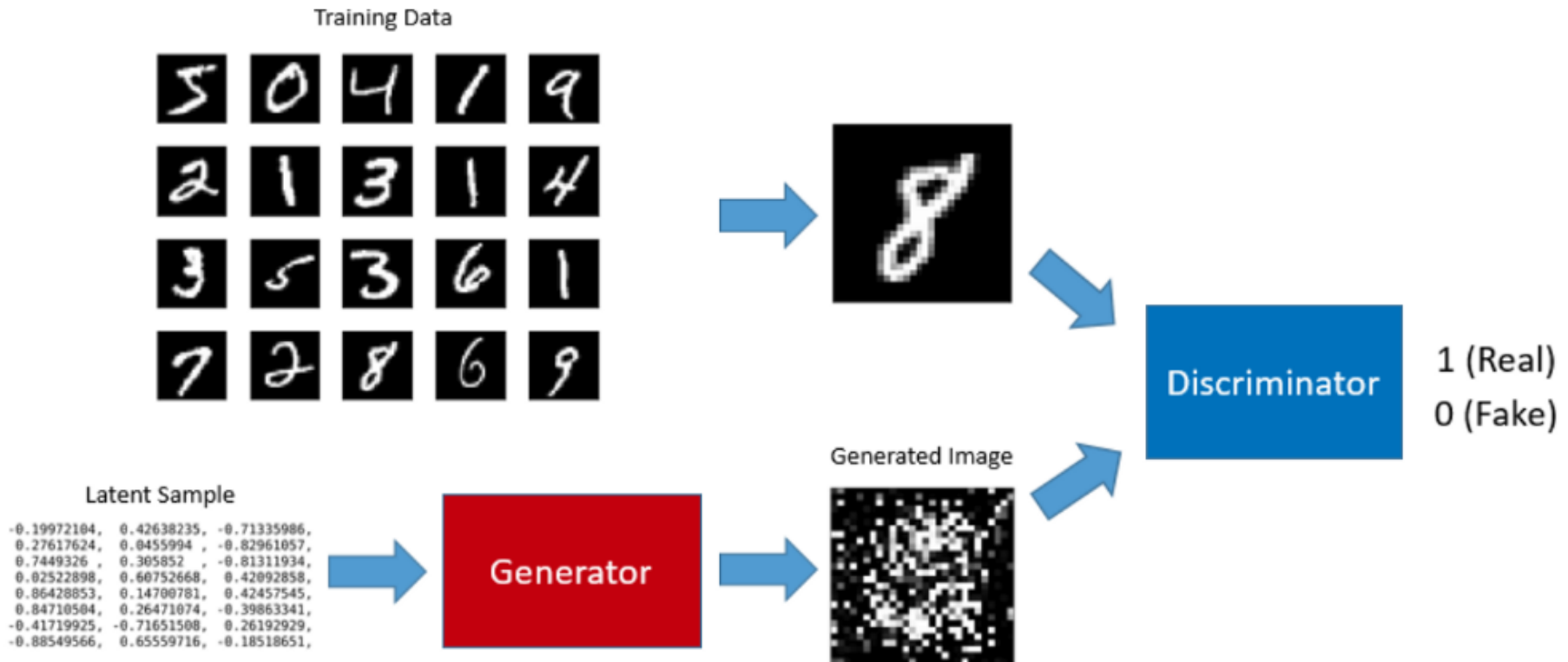
Generative Adversarial Neural Networks

One need to set two networks: Generator and a Discriminator



Generative Adversarial Neural Networks

The key here is the training There is a loop:
first we train the discriminator. Let the generator create some images out of noise and train **the discriminator only** to define true or false.



Generative Adversarial Neural Networks

The key here is the training There is a loop:
first we train the discriminator. Let the generator create some images out of noise and train **the discriminator only** to define true or false. Consider a Keras model D to be the discriminator, G the Generator and AD the Adversarial Model, i.e., the full model [G,D]

```
generated_images = G.predict(x=random_data, batch_size=batch_size)

d_loss_real = D.train_on_batch(image_real_batch, output_true)
d_loss_fake = D.train_on_batch(generated_images, output_false)
d_loss = 0.5 * np.add(d_loss_fake, d_loss_real)
```

Generative Adversarial Neural Networks

The key here is the training

Then we set the Discriminator to be **non-trainable**. After that we optimize the generator. We create some random noise data flows through the Network (Generator and Discriminator). The Generator weights will be updated by backpropagation from the non-trainable Discriminator to the Generator.

```
D.trainable = False
y = np.ones([batch_size, 1])
noise = np.random.uniform(-1.0, 1.0, size=[batch_size, 100])
AD_loss = AD.train_on_batch(random_data, y)
D.trainable = True
```



Generative Adversarial Neural Networks are hard to train...

Some general Advice

Gain is particularly sensible to vanishing gradients/ sparse gradients

LeakyReLU is recommended instead of ReLU.

If Discriminator loss converges rapidly and thus prevents the Generator from learning make its learning rate bigger than the Adversarial model learning rate you can also use a different training noise sample for the Generator.

If generated images look like noise. Use dropout on both Discriminator and Generator. Low dropout values generate more realistic images.



Some Scary smiles from DCGAN



Cycle GAN: Make your own Monet

[arXiv:1703.10593](https://arxiv.org/abs/1703.10593)



Monet $\leftrightarrow$ Photos



Monet $\rightarrow$ photo



photo $\rightarrow$ Monet

Summer $\leftrightarrow$ Winter



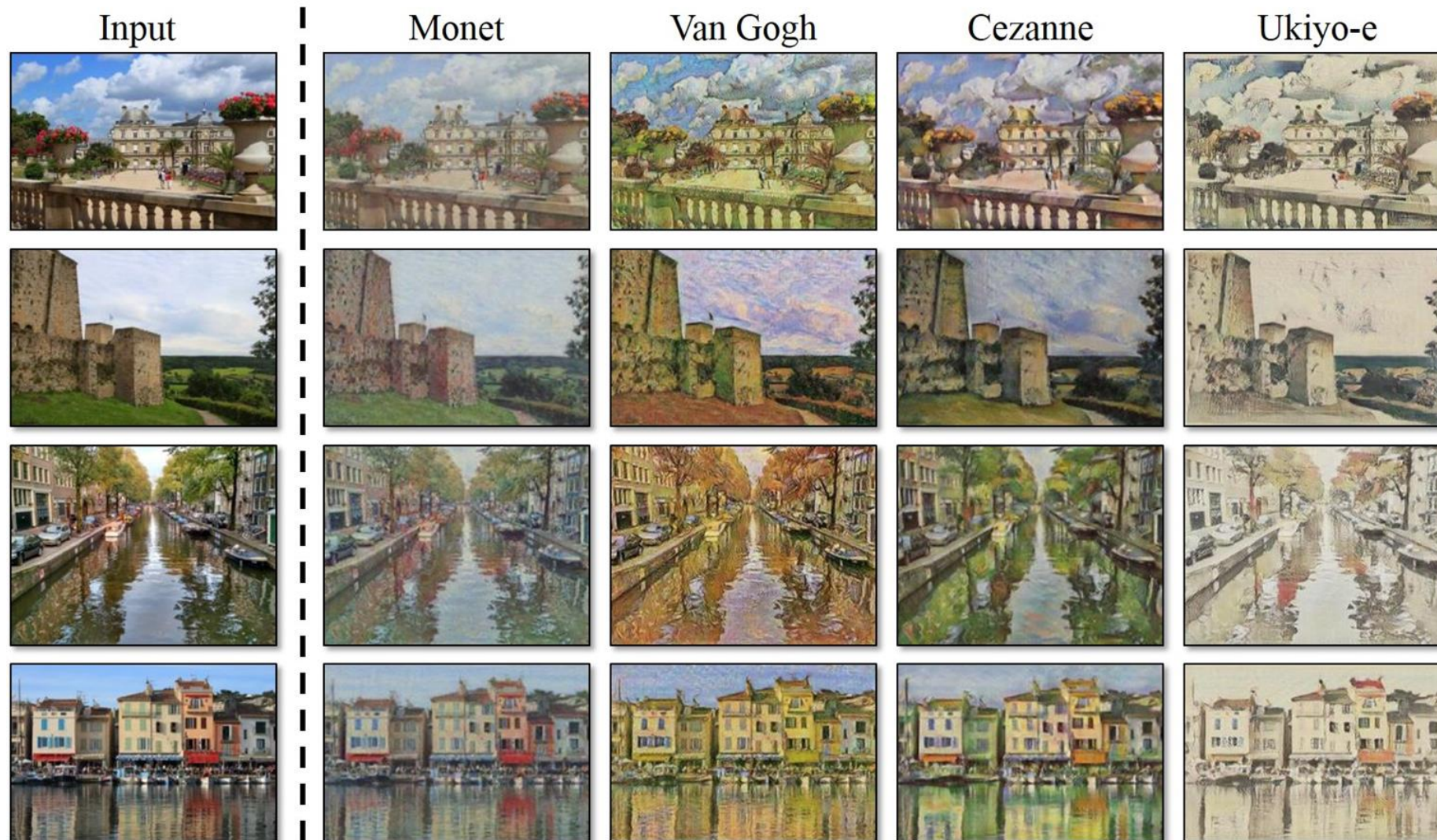
summer $\rightarrow$ winter



winter $\rightarrow$ summer

Cycle GAN: Make your own Monet

[arXiv:1703.10593](https://arxiv.org/abs/1703.10593)



Reinforcement Learning

If we set the discount factor $\gamma=0$, then our strategy will be short-sighted and we rely only on the immediate rewards. If we want to balance between immediate and future rewards, we should set discount factor to something like $\gamma=0.9$. If our environment is deterministic and the same actions always result in same rewards, then we can set discount factor $\gamma=1$.

A good strategy for an agent would be to always choose an action that maximizes the (discounted) future reward.

Discounted Future Reward

$$R_t = r_t + \gamma r_{t+1} + \gamma^2 r_{t+2} \dots + \gamma^{n-t} r_n$$

$$R_t = r_t + \gamma(r_{t+1} + \gamma(r_{t+2} + \dots)) = r_t + \gamma R_{t+1}$$

Q Learning

The maximum discounted future reward when we perform action a in state s , and continue optimally from that point on

The best possible score at the end of the game after performing action a in state s

$$Q(s_t, a_t) = \max R_{t+1}$$

Bellman equation

$$Q(s, a) = r + \gamma \max_{a'} Q(s', a')$$

maximum future reward for this state and action is the immediate reward plus maximum future reward for the next state

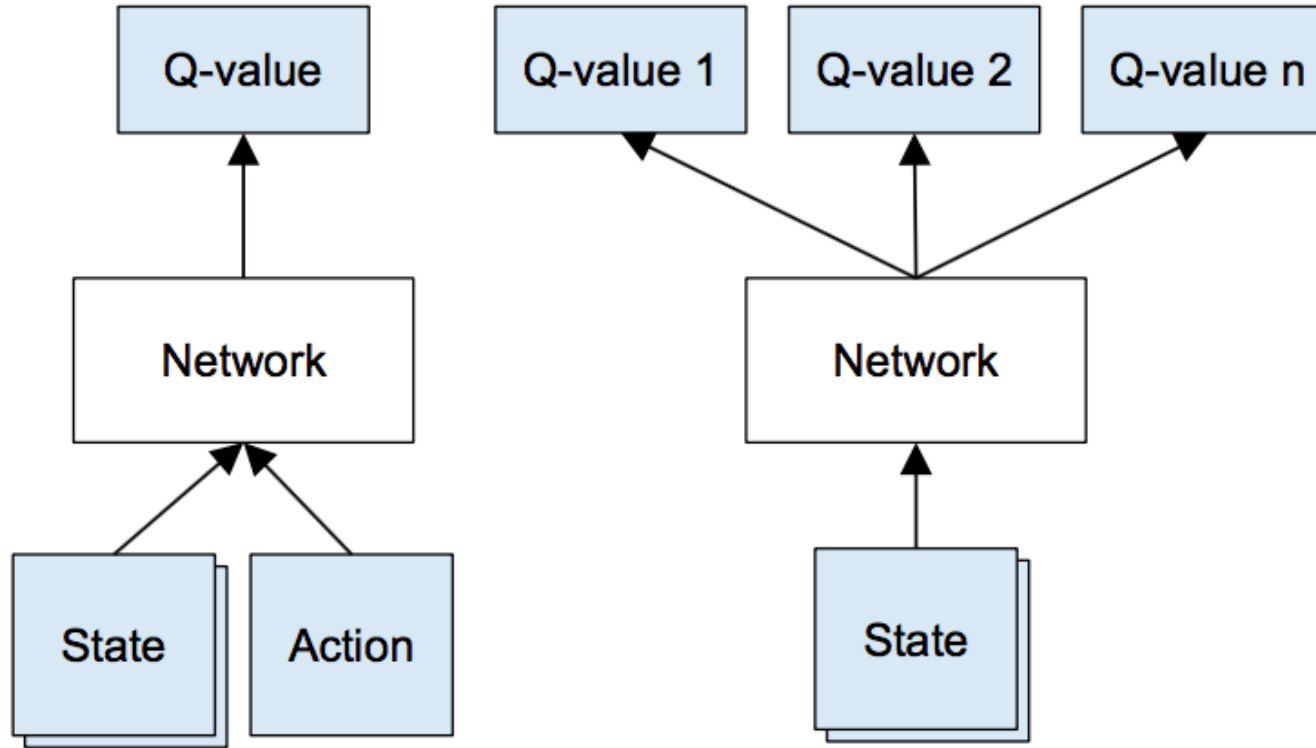
Q Learning

$$L = \frac{1}{2} \left[\underbrace{r + \max_{a'} Q(s', a')}_{\text{target}} - \underbrace{Q(s, a)}_{\text{prediction}} \right]^2$$

```
initialize  $Q[num\_states, num\_actions]$  arbitrarily
observe initial state  $s$ 
repeat
    select and carry out an action  $a$ 
    observe reward  $r$  and new state  $s'$ 
     $Q[s, a] = Q[s, a] + \alpha(r + \gamma \max_{a'} Q[s', a'] - Q[s, a])$ 
     $s = s'$ 
until terminated
```

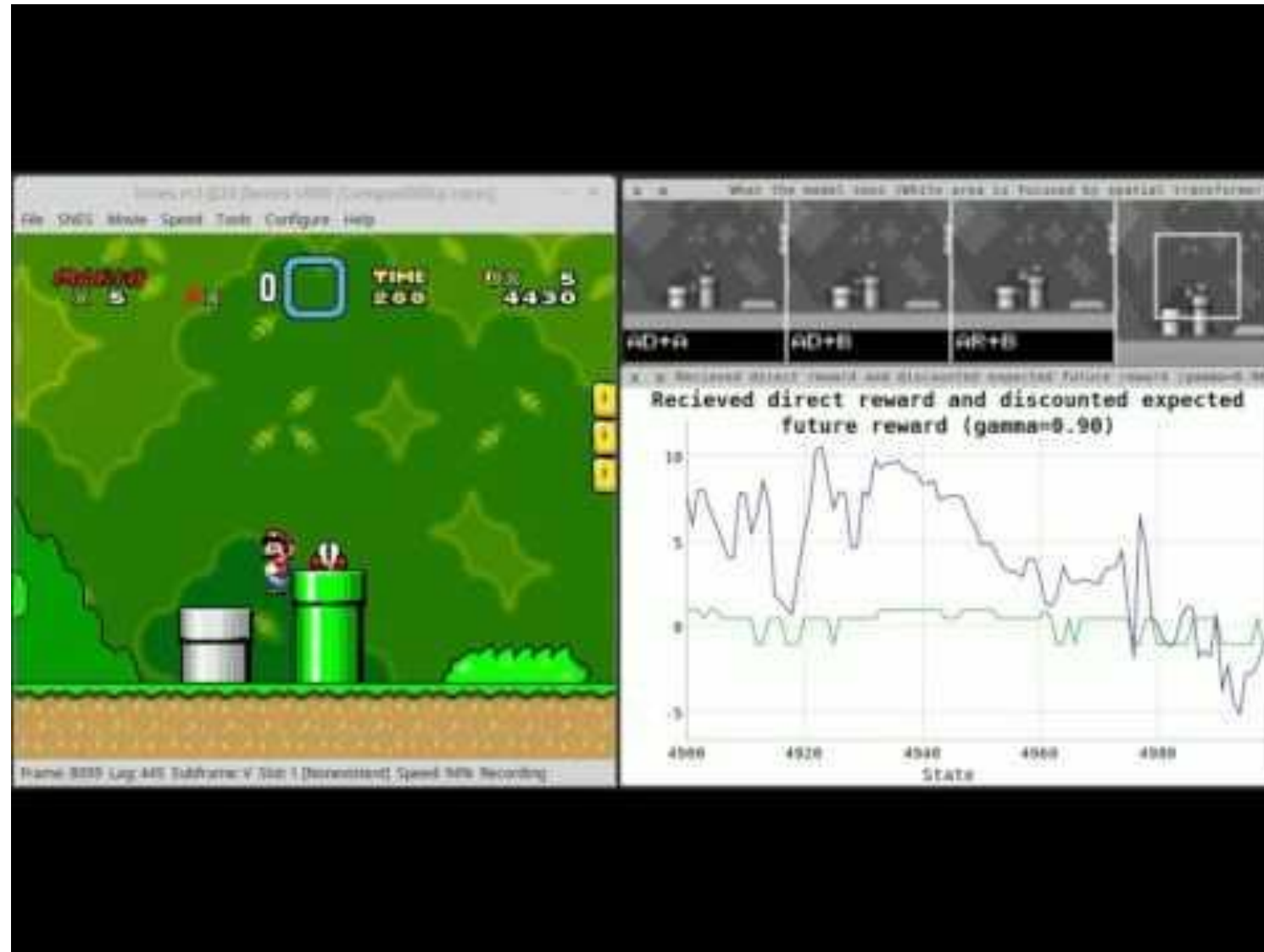
Reinforcement Learning

Play video Games!

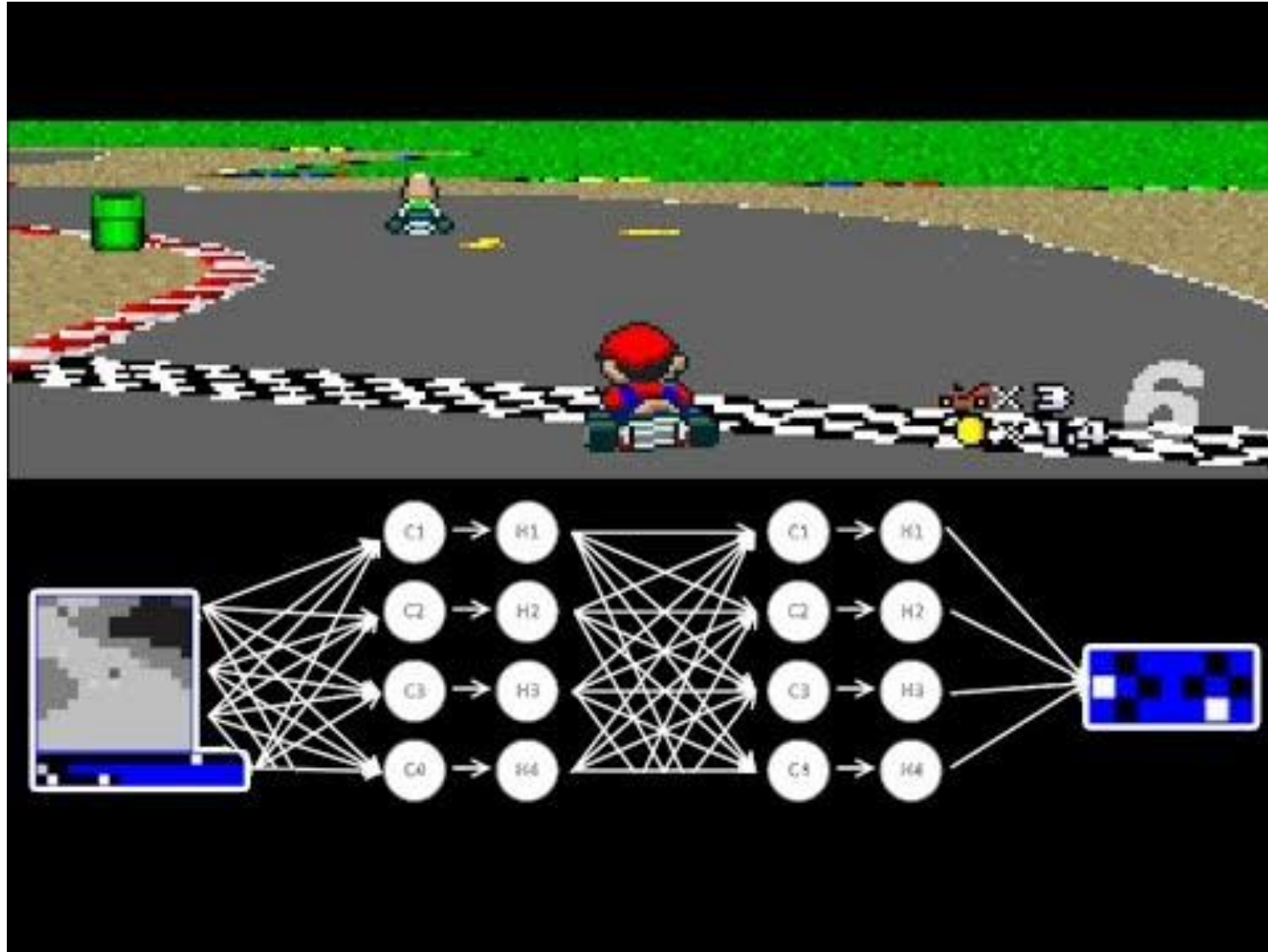


We could also take only game states input and output the Q-value for each possible action.

Let's Play!



Let's Play!



Imaging Atmospheric Cherenkov Telescope Layout

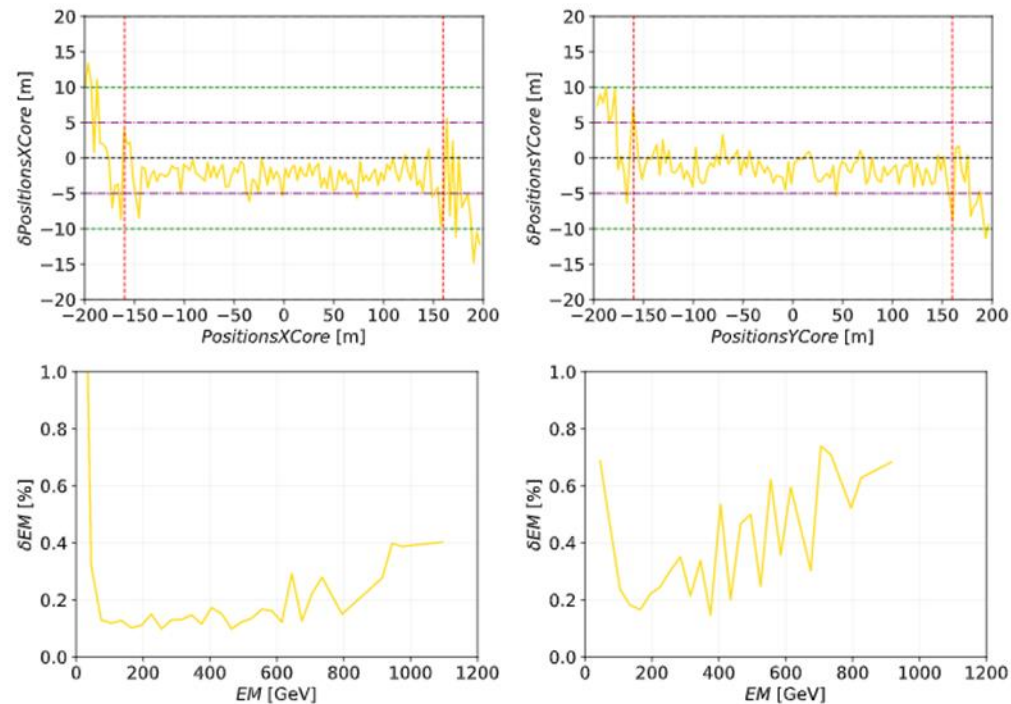
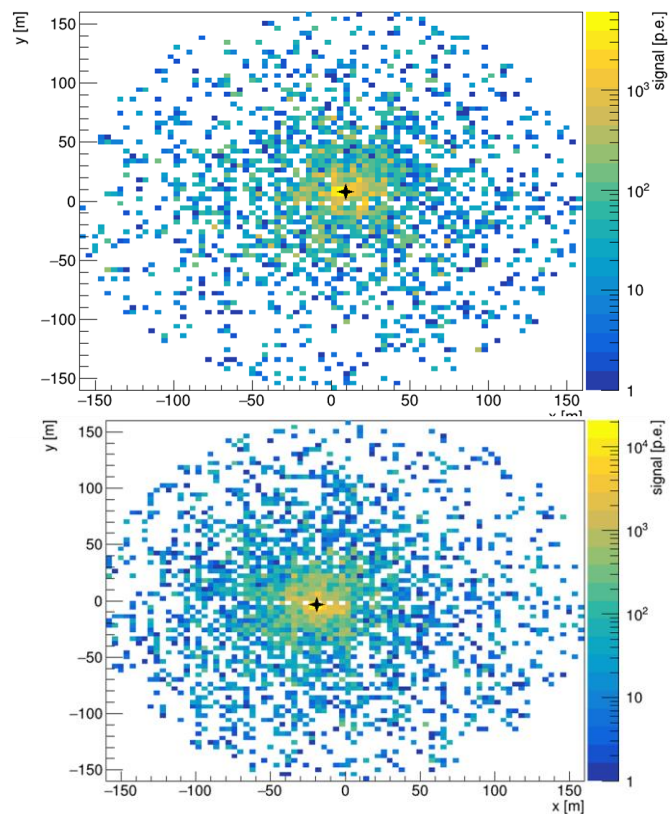
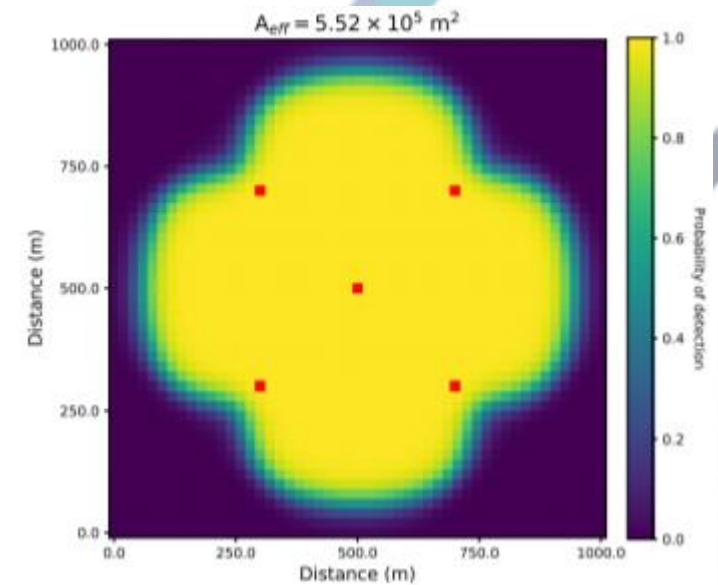
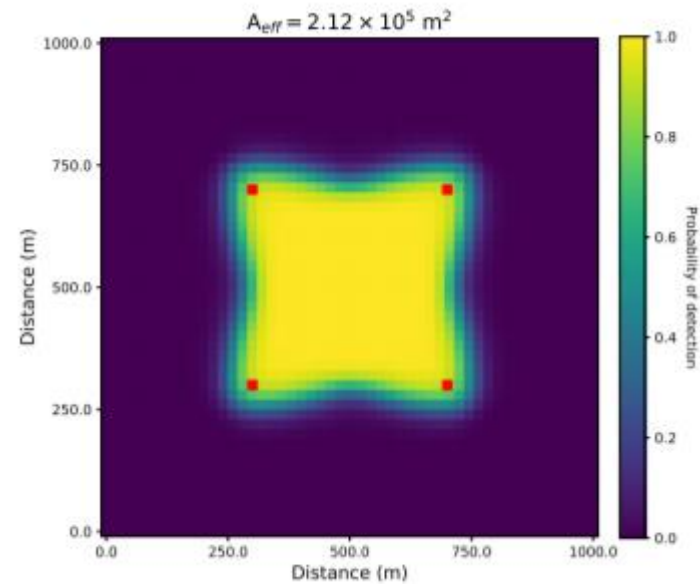
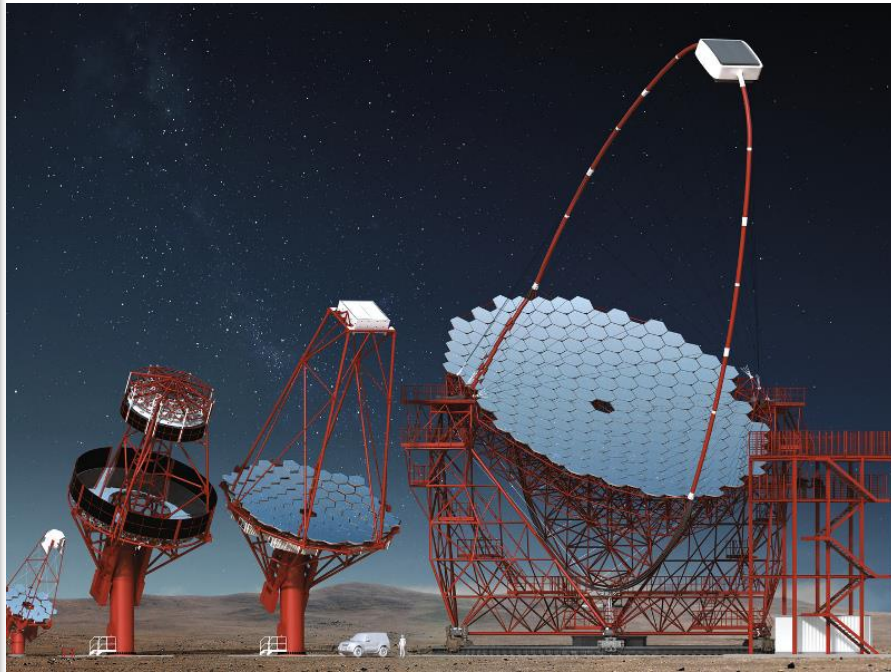


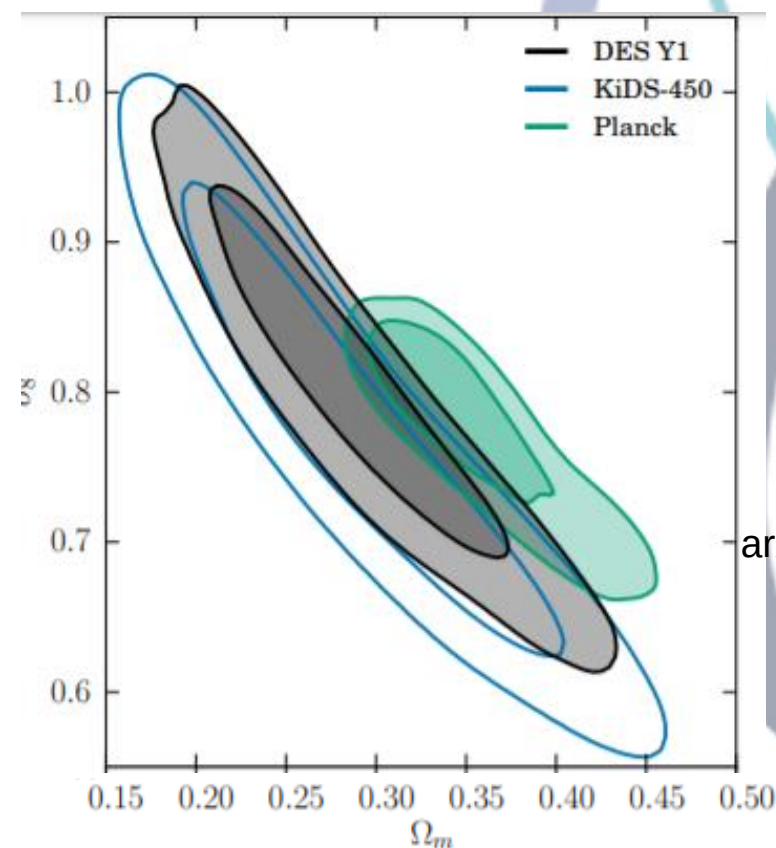
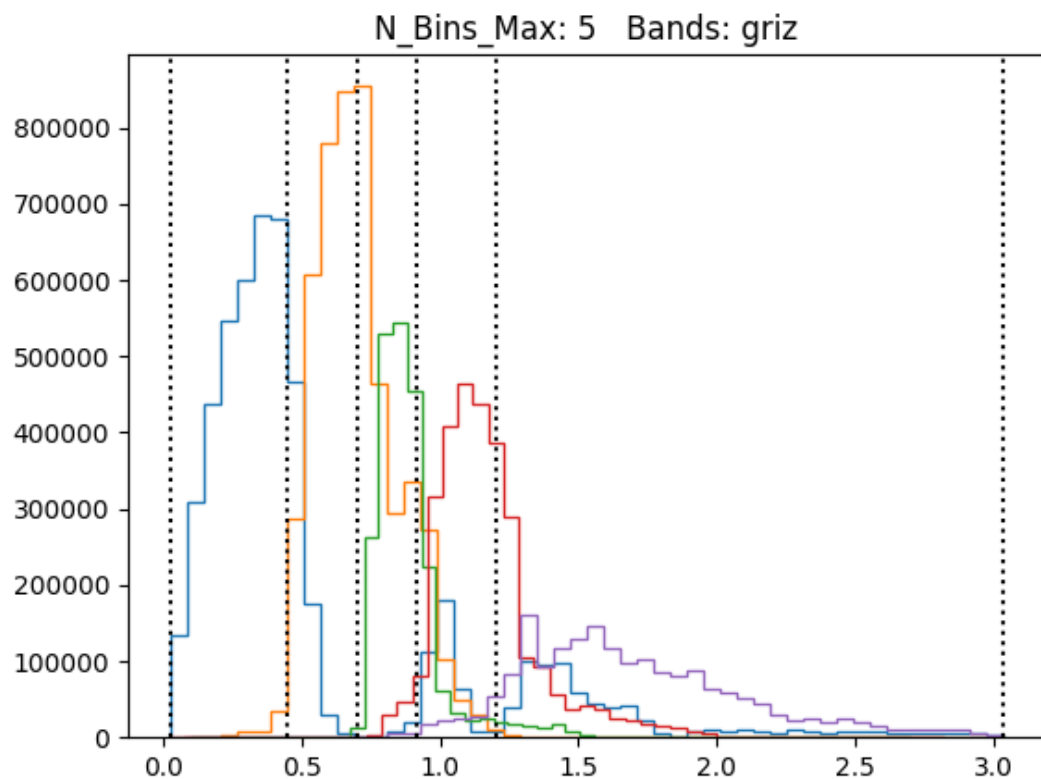
Figure 2: The panels summarize the results using a Bayesian Deep Neural Network using EfficientNet B0 architecture. The upper panels present the X_{core} (left) and Y_{core} (right) bias. The lower panels present the ground energy bias (%) for the region near the detector center (left) and outside detector range (right).

Imaging Atmospheric Cherenkov Telescope Layout



Redshift Bias Challenge

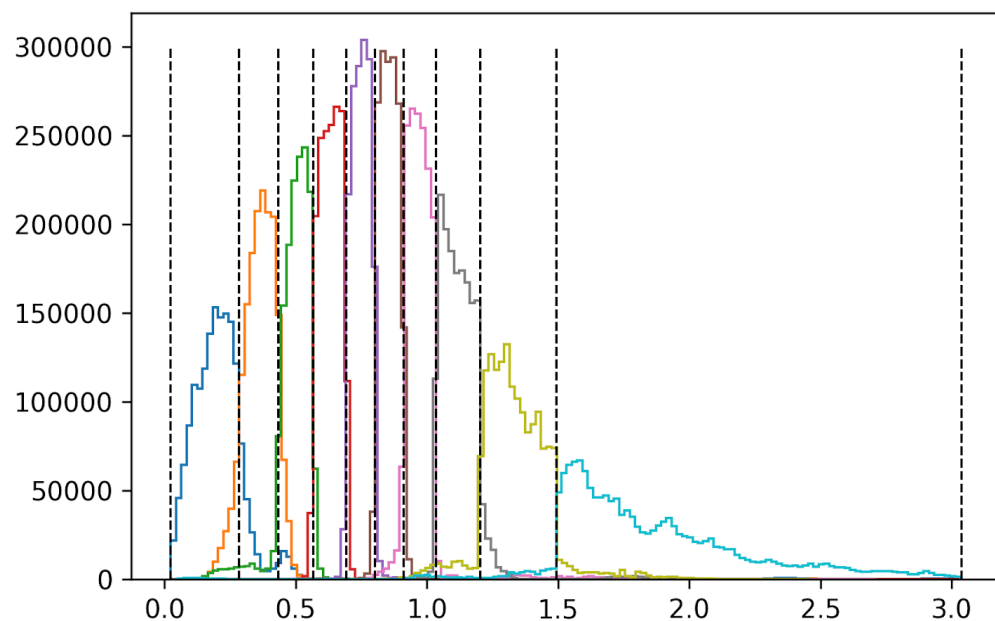
In this challenge you are asked to group galaxies into tomographic bins using only the quantities we generate using the metacalibration method. These quantities are the only ones for which we can compute a shear bias correction associated with the division.



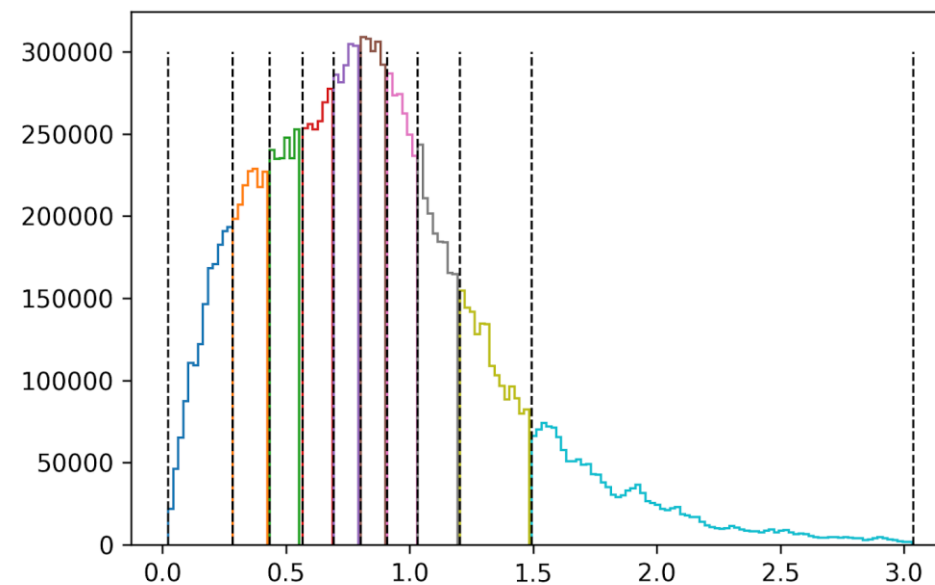
arxiv:1708.01538

Redshift Bias Challenge

In this challenge you are asked to group galaxies into tomographic bins using only the photometry only.



LSTM model with AutoML

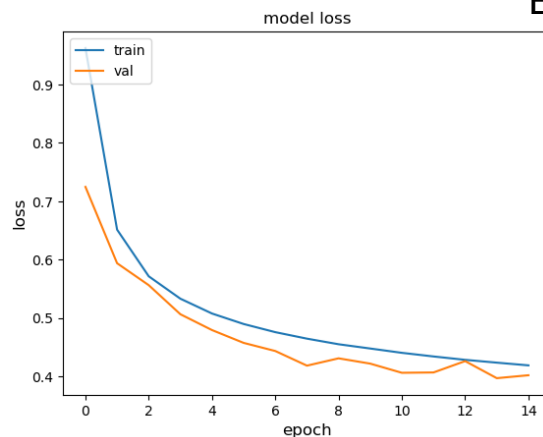
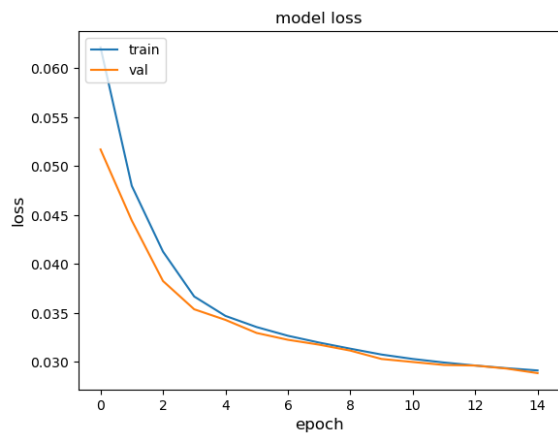


Truth Table

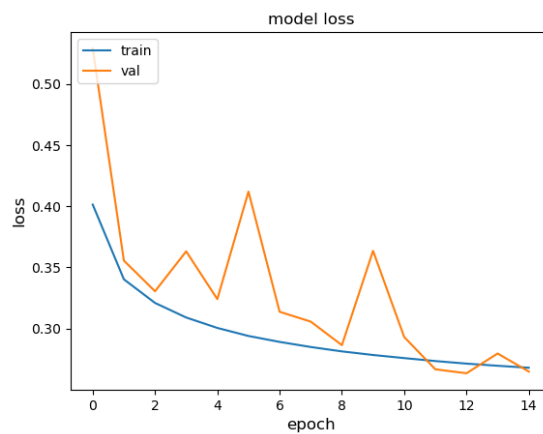
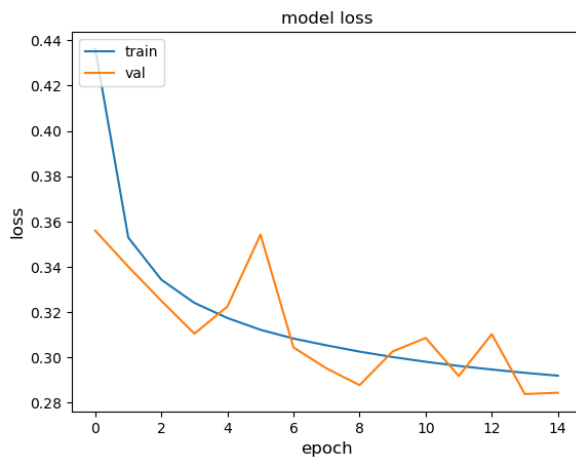
A Deep Ensemble

In this challenge you are asked to group galaxies into tomographic bins using only the quantities we generate using the metacalibration method. These quantities are the only ones for which we can compute a shear bias correction associated with the division.

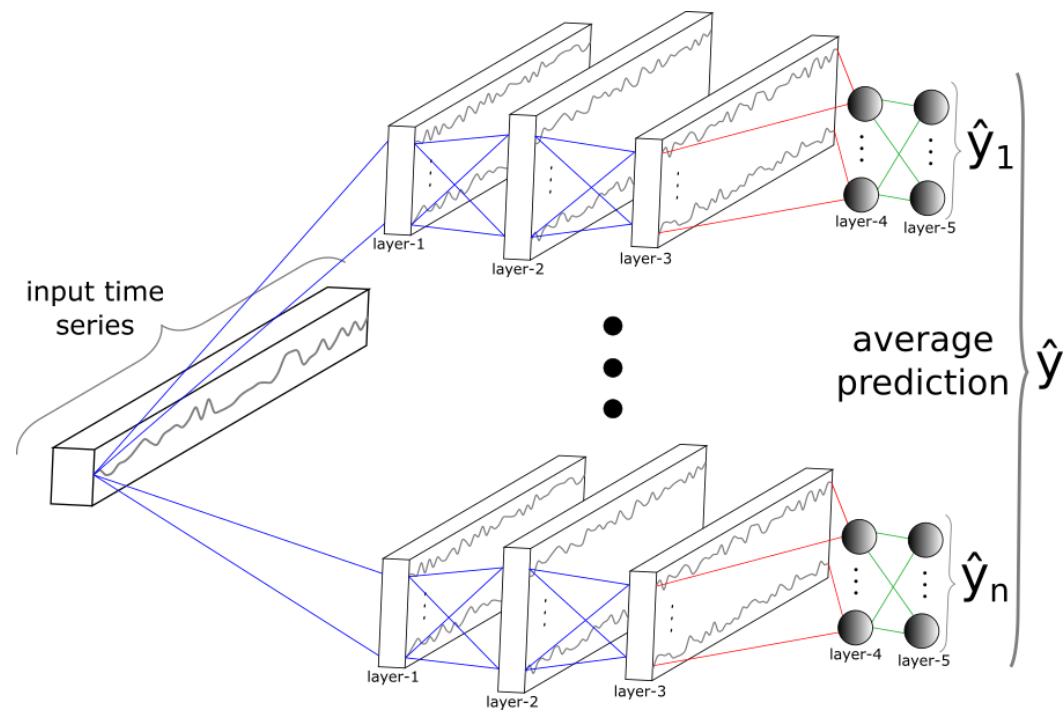
CNN-1D



FCCNN



Encoder

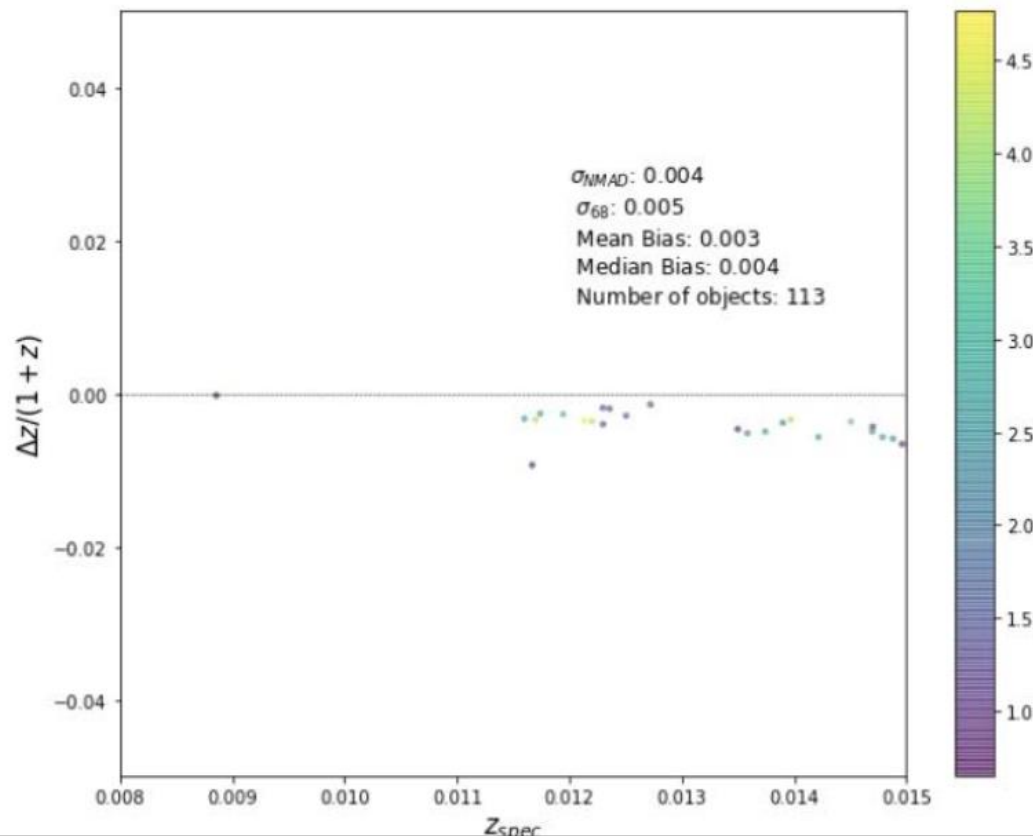


Resnet

Precise photo-zs at low redshift regime

The S-PLUs deals with objects at low- z as nearby galaxy clusters, e.g. fornax cluster at $z \sim 0.005$. At this limit the photo- z errors are usually higher than the actual photo- z . Peculiar velocity also plays an important role here. The advantage here is the use of narrow band filters (12 filters)

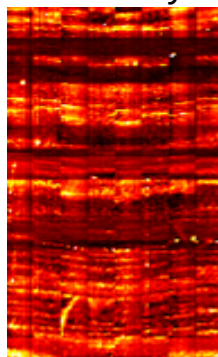
The strategy is to develop an Ensemble of Bayesian Nets (outputs PDFs), rely on huge data augmentation and to optimize it for alternative loss that penalizes heavily the errorbars.



PRELIMINARY

resistivity

ultrasonic



Journal of Petroleum Science and Engineering 170 (2018) 315–330

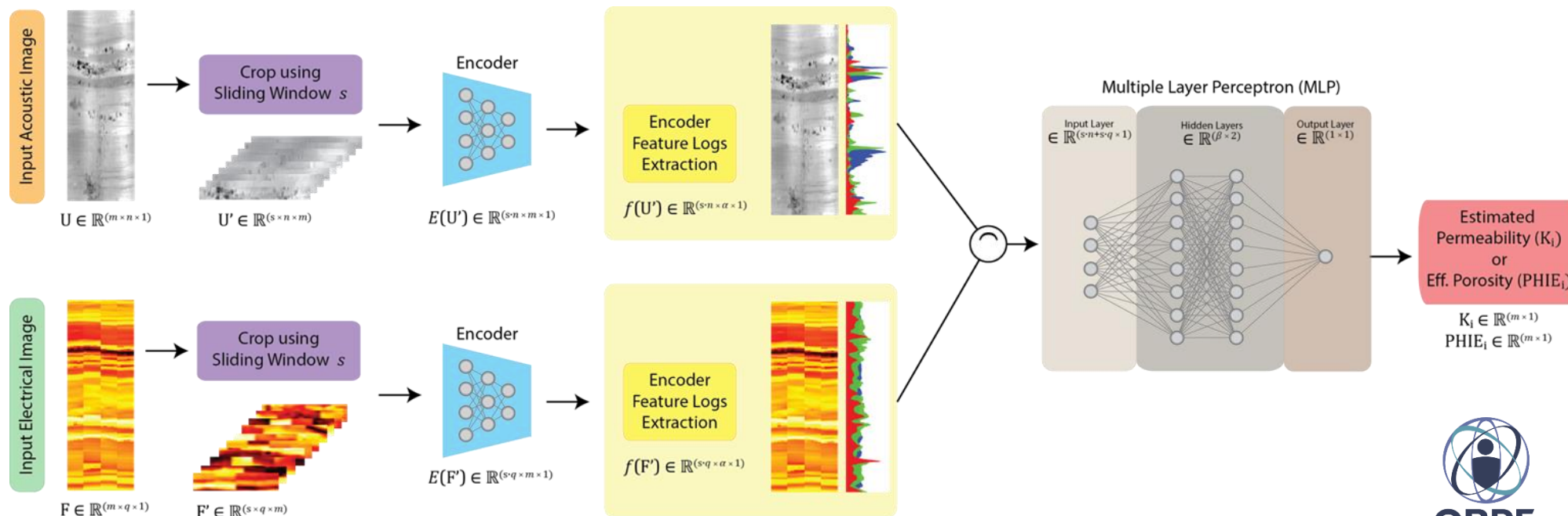
Estimation of permeability and effective porosity logs using deep autoencoders in borehole image logs from the brazilian pre-salt carbonate

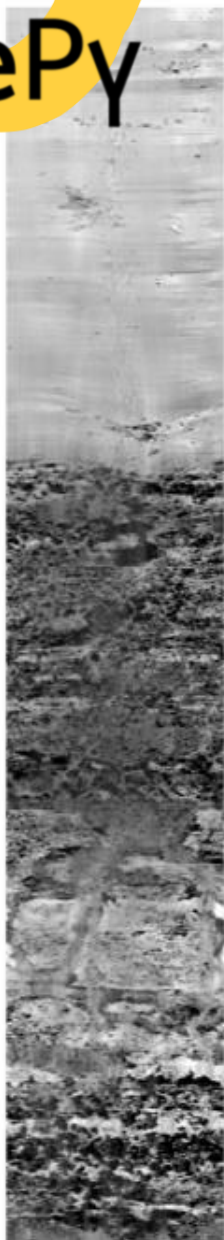
Manuel Blanco Valentín^{a,*}, Clécio R. Bom^b, André Luiz Martins Compan^c, Maury Duarte Correia^c, Candida Menezes de Jesus^d, Anelise de Lima Souza^d, Márcio P. de Albuquerque^a, Marcelo P. de Albuquerque^a, Elisângela L. Faria^a

<https://doi.org/10.1016/j.petrol.2018.06.038>

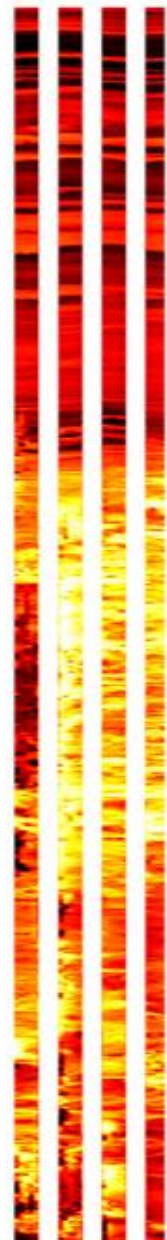


Deep Learning technique to estimate perm/porosity along the borehole using a combination of enhanced ultrasound and resistivity images

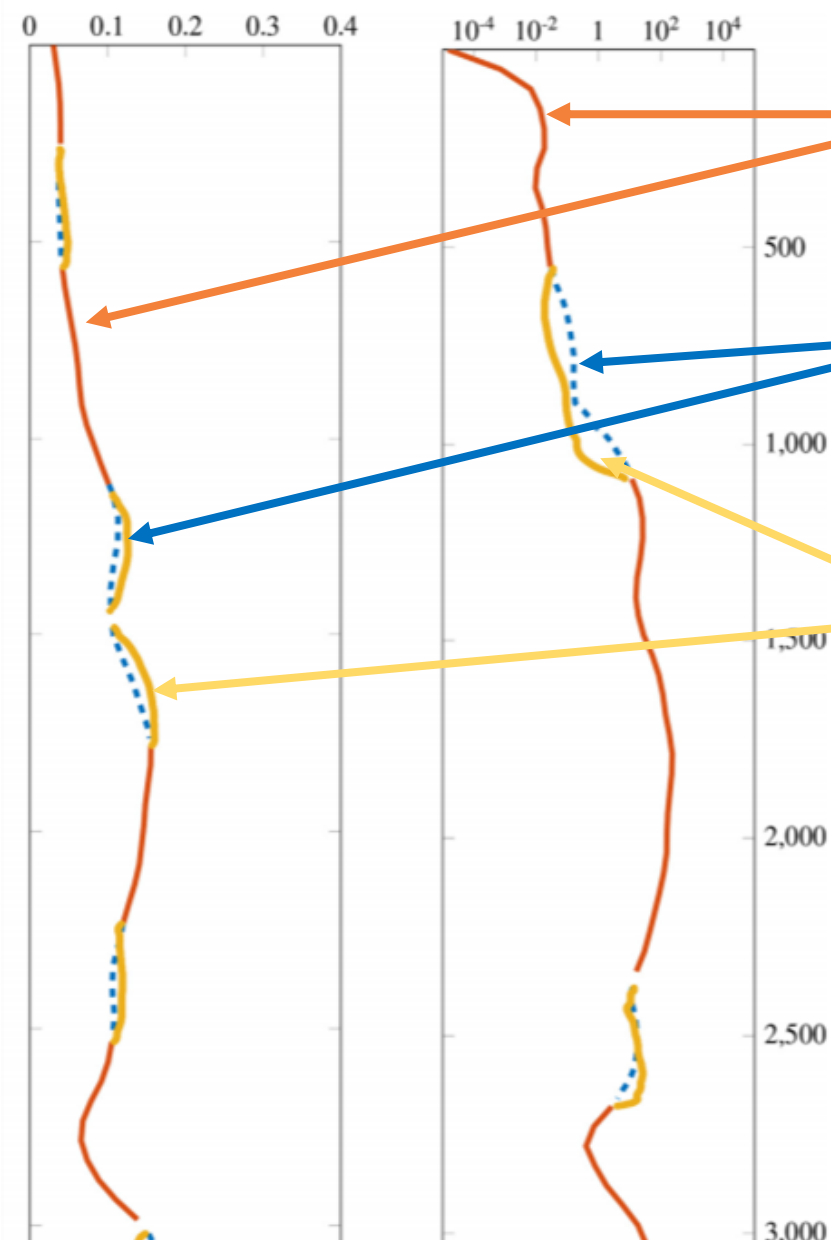




ultrasonic



presistivity



Effective
porosity

Formation
permeability

Orange lines:

Original values used for training the net

Blue lines:

Original values used for testing the net

Yellow lines:

Predicted values by the net



Comparison to other methods

Table 2

Comparison between our best result for the Estimation of effective porosity measurements and the results of previous authors.

	R^2 (%)	Data Used	Method
Hamada and Elshafei (2010)	66.00	RHOB ^a , GR ^b , NPHI ^d , DRT/SRT ^e , NMR (T2) ^f	FFNN ^c
Our method	96.30	UBI/OBMI ^g	sAE ^h + SVM ⁱ

Highlights:

We were the only ones to retrieve permeability using automated method in a quantitative comparable approach

Other groups did not used images (except Menezes de Jesus et al.)

The other methods that estimated porosity were not validated in pre-salt.



Comparison to other methods

Table 3

Comparison between our best result for the Estimation of permeability measurements and the results of previous authors.

	R^2 (%)	Data used	Method
Mohaghegh et al. (1995)	96.30	Depth, GR ⁹ , RHOB ⁸ , DRT ¹² , ZSS ^b	ANN ^a
Hamada and Elshafei (2010)	93.75	RHOB ⁸ , GR ⁹ , NPHI ¹¹ , DRT/SRT ¹² , NMR (T2) ¹³	FFNN ¹⁰
Abdideh (2012)	89.00	Depth, CDT ^c , DT ^d , NPHI ¹¹ , RHOB ⁸ , PHIT ^e , WS ^f , GR ⁹	FFNN ¹⁰
Elkattatny et al. (2017)	95.00	RHOB ⁸ , GR ⁹ , NPHI ¹¹ , DT ^d , DRT/SRT ¹² , WS ^f	BNN ^g + ANFIS ^h , SVM ¹⁶
Our method	96.06	Caliper, PHIT ^e , UBI/GBM ¹⁴	sAE ¹⁵ + SVM ¹⁶



But... it is Sandstone

Highlights:

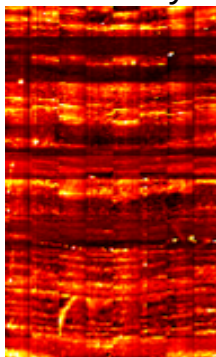
Other groups did not use images (except Menezes de Jesus et al.)

The other methods that estimated porosity were not validated in pre-salt.



Pre-salt Carbonate

resistivity

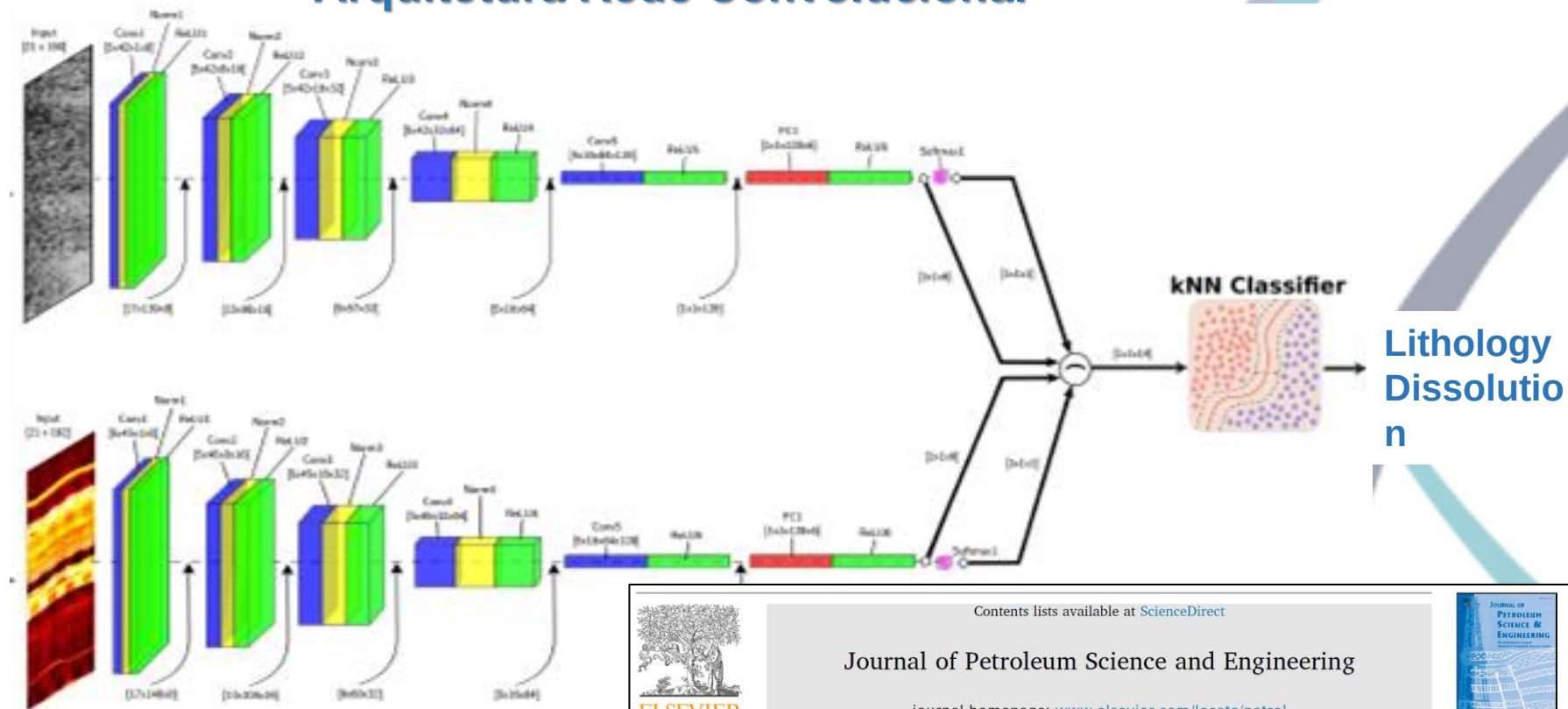


ultrasonic

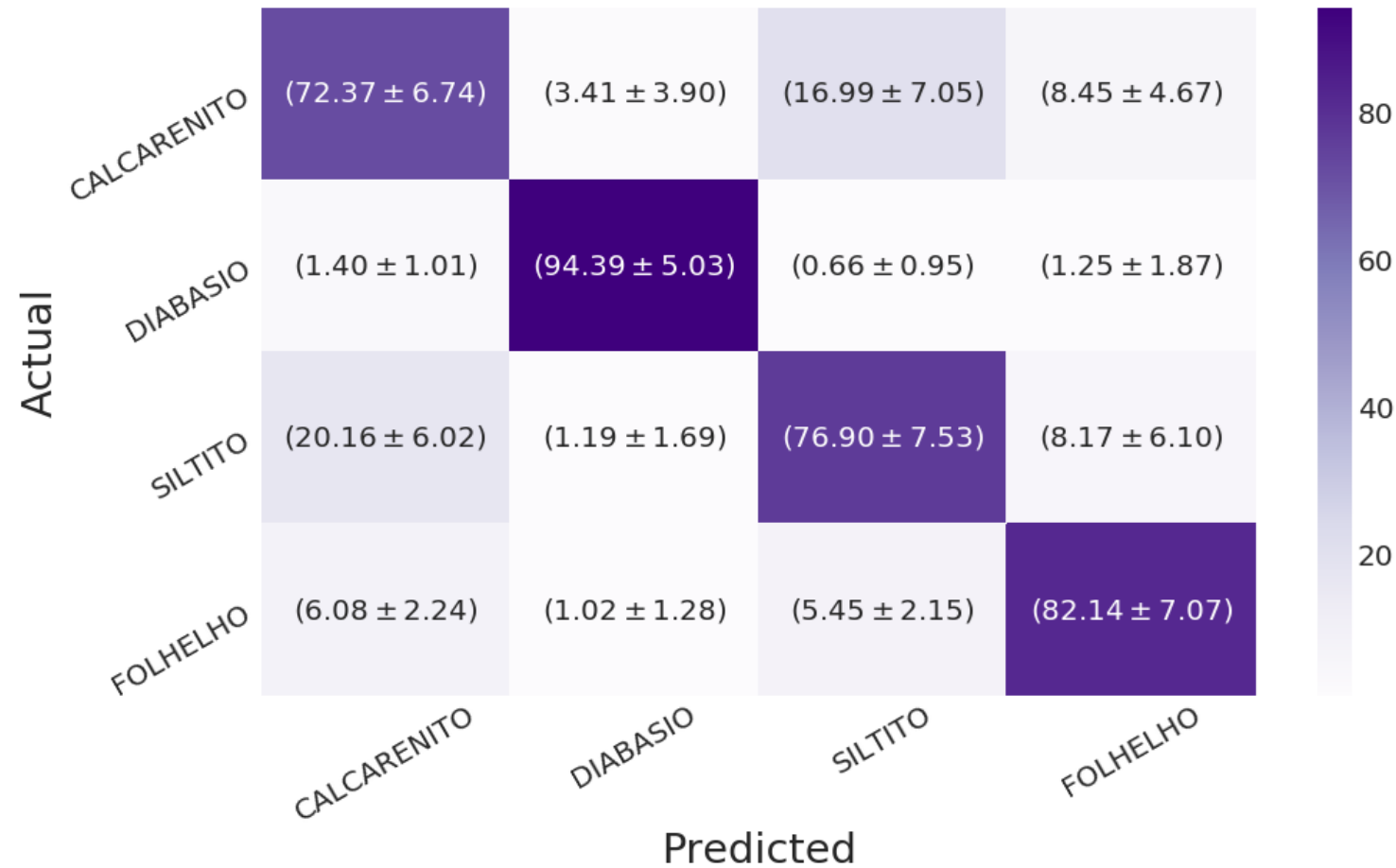


Deep Learning architecture to make a fast and reliable analysis from the Ultrasound and Resistivity images that can be validated by the geologist.

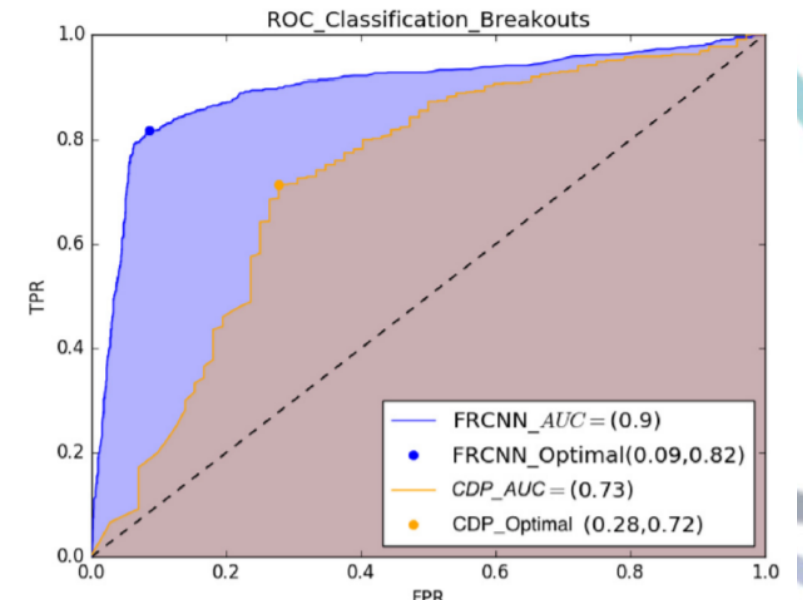
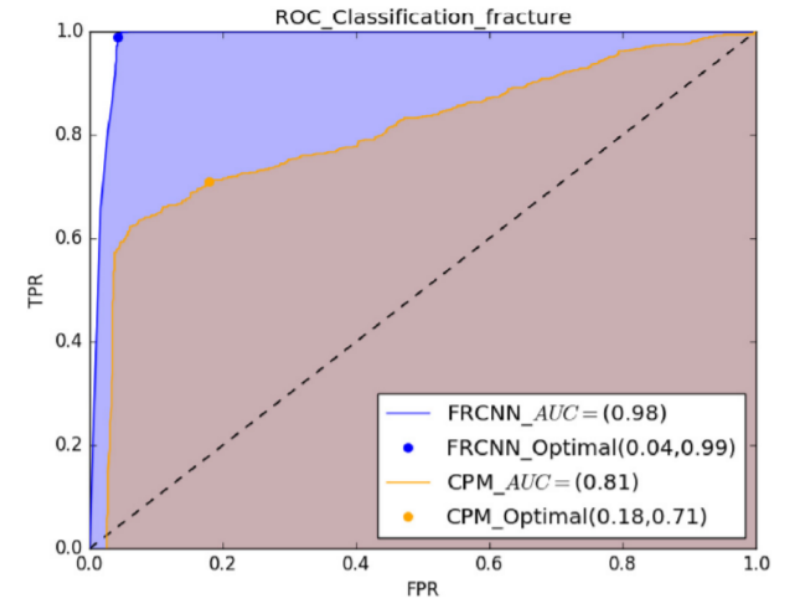
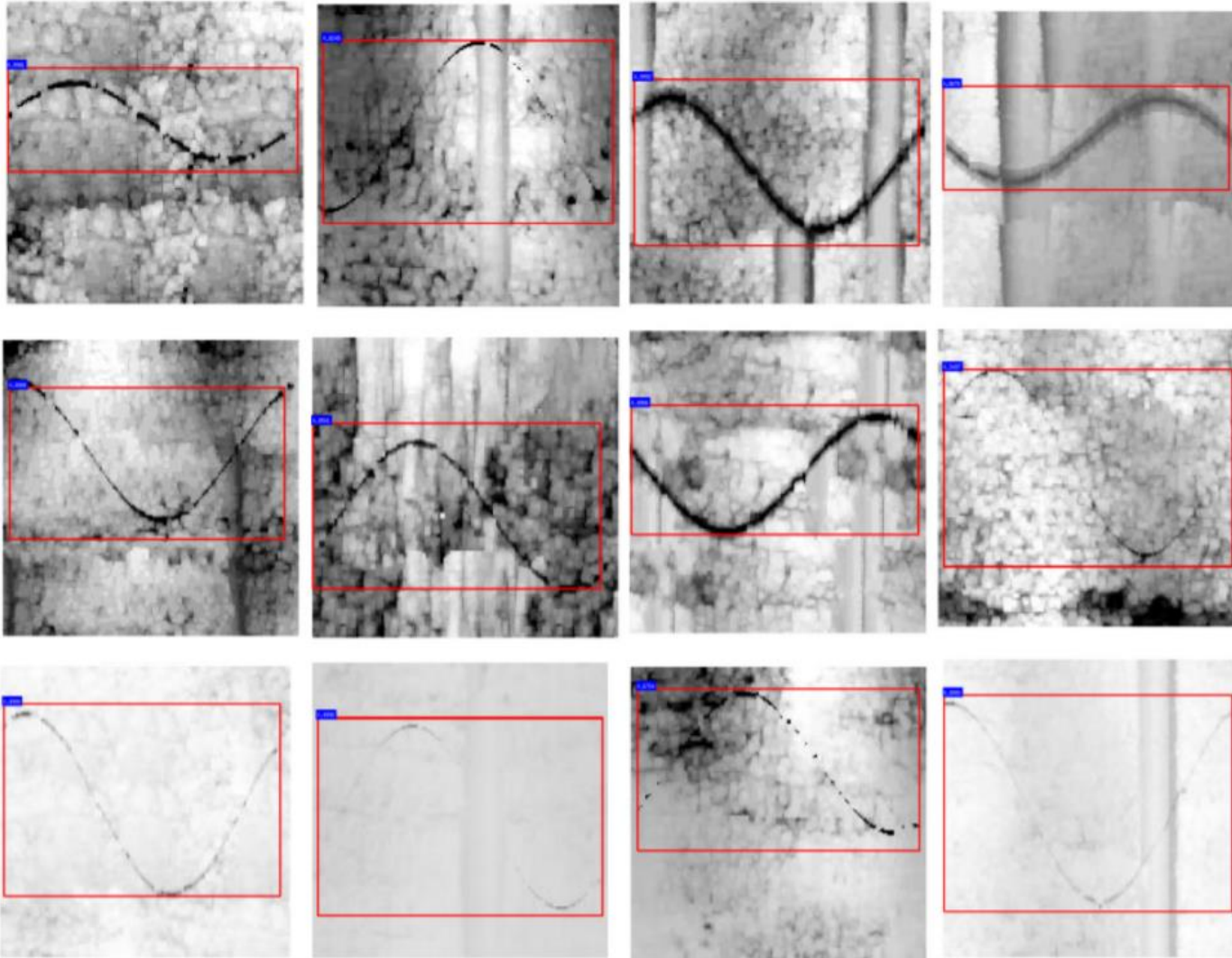
Arquitetura Rede Convolutacional



Lithology Classification in pre-salt



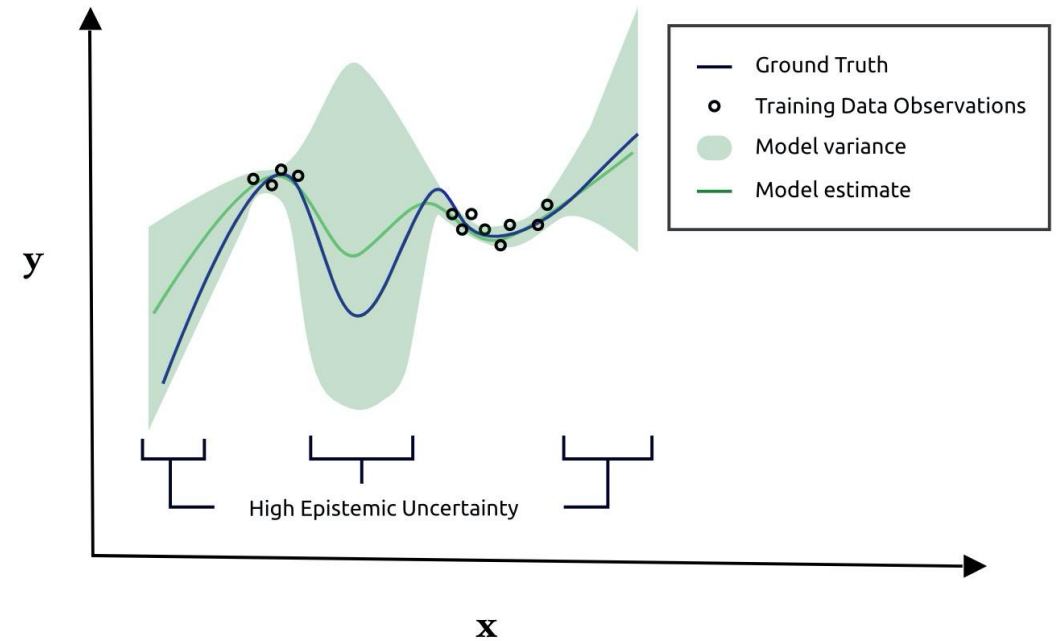
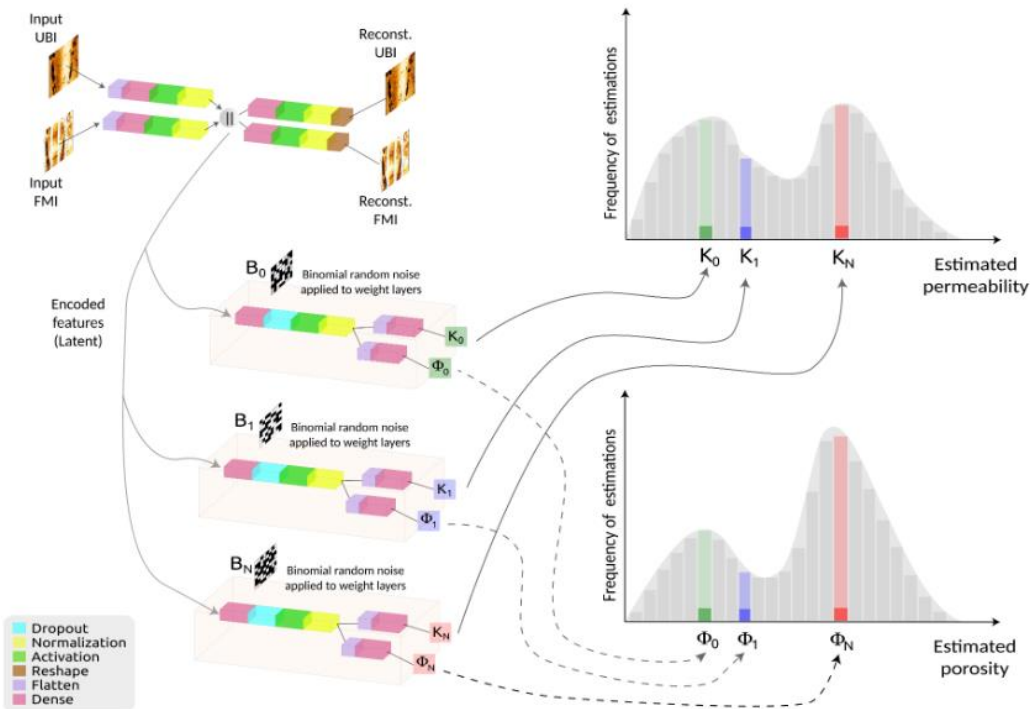
Automatic Fractures and Breakouts detection with Deep Learning



Data Driven Deep Learning Bayesian Inversing modelling

DL Models are Functions Universal Approximators. Sampling over the weights is equivalent to sampling over a space of functions as in a gaussian process.

$$p(\mathbf{y}|\mathbf{x}, \mathbf{X}, \mathbf{Y}) = \int p(\mathbf{y}|\mathbf{x}, \omega) p(\omega|\mathbf{X}, \mathbf{Y}) d\omega.$$



It allows us to derive fast multidimensional probability density estimation in a exchangeable Bayesian inference.



Bayesian deep networks for absolute permeability and porosity uncertainty prediction from image borehole logs from brazilian carbonate reservoirs

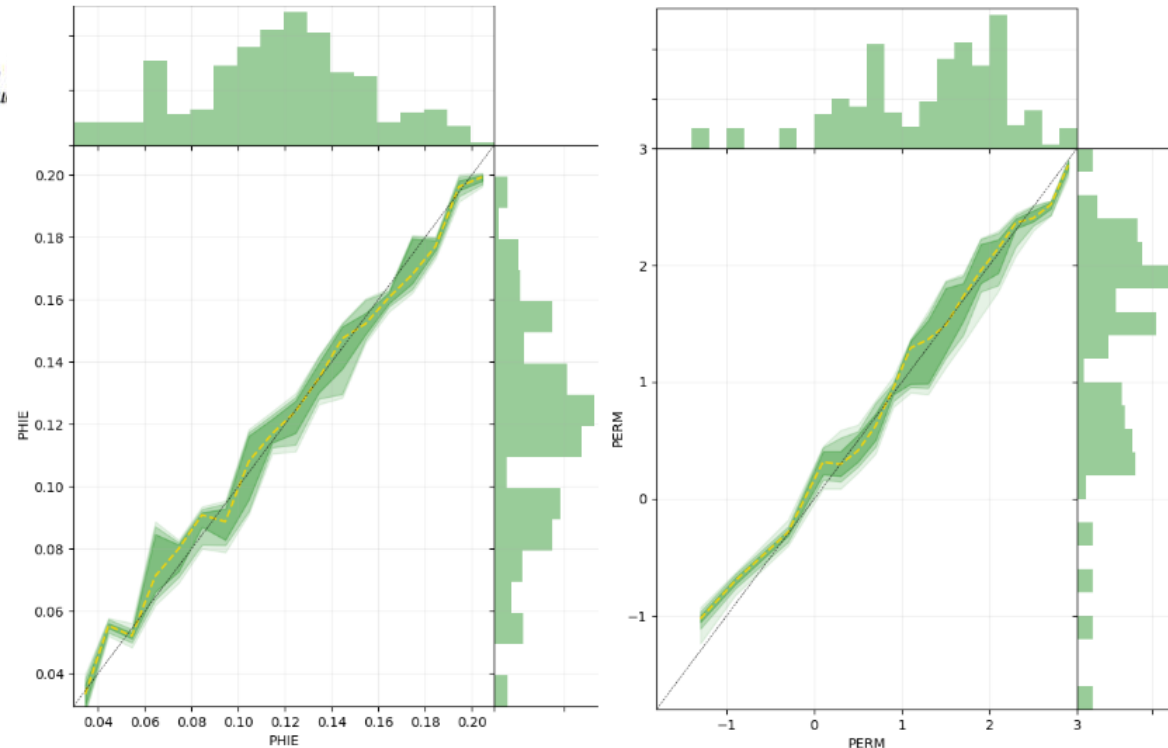
Clécio R. Bom^{a,b,*}, Manuel Blanco Valentín^a, Bernardo M.O. Fraga^a, Jorge Campos^c, Bernardo Coutinho^c, Luciana O. Dias^a, Elisangela L. Faria^a, Márcio P. de Albuquerque^a, Marcelo P. de Albuquerque^a, Maury Duarte Correia^c

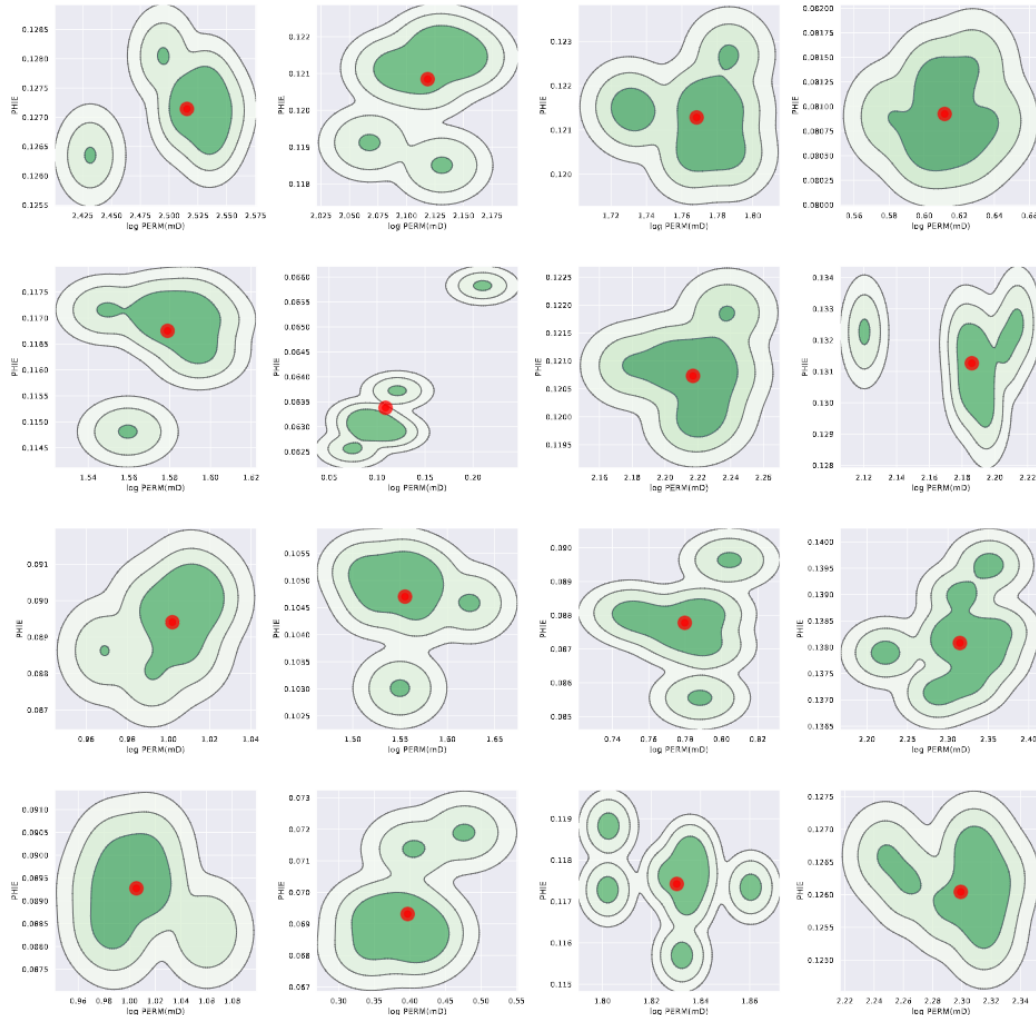
^a Centro Brasileiro de Pesquisas Físicas (CBPF) - Rua Xavier Sigaud, 150, Ed. César Lattes, Urca - Rio de Janeiro,

^b Centro Federal de Educação Tecnológica Celso Suckow da Fonseca, Rodovia Márcio Covas, lote, quadra J - Itaguaí,

^c Centro de Pesquisas e Desenvolvimento Leopoldo Américo Miguez de Mello, Brazil

DL Models are Functions Universal Approximators. Sampling over the weights is equivalent to sampling over a space of functions as in a gaussian process.





$R^2(\%)$ Perm	$R^2(\%)$ Phie
99.63 ± 0.31	99.31 ± 0.92
99.73 ± 0.18	99.71 ± 0.18

In a Pre-salt Carbonate, compared to plug calibrated model
No human intervention!



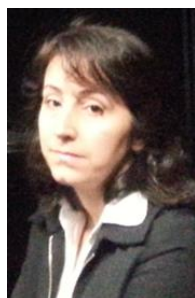


Inteligência artificial e aplicações em física

Clécio Roque De Bom – debom@cbpf.br



clearnightsrthebest.com



Elisangela L. Faria
(CBPF)

