On Lindenmayer Systems and Autoencoders

Andrej Lúčny

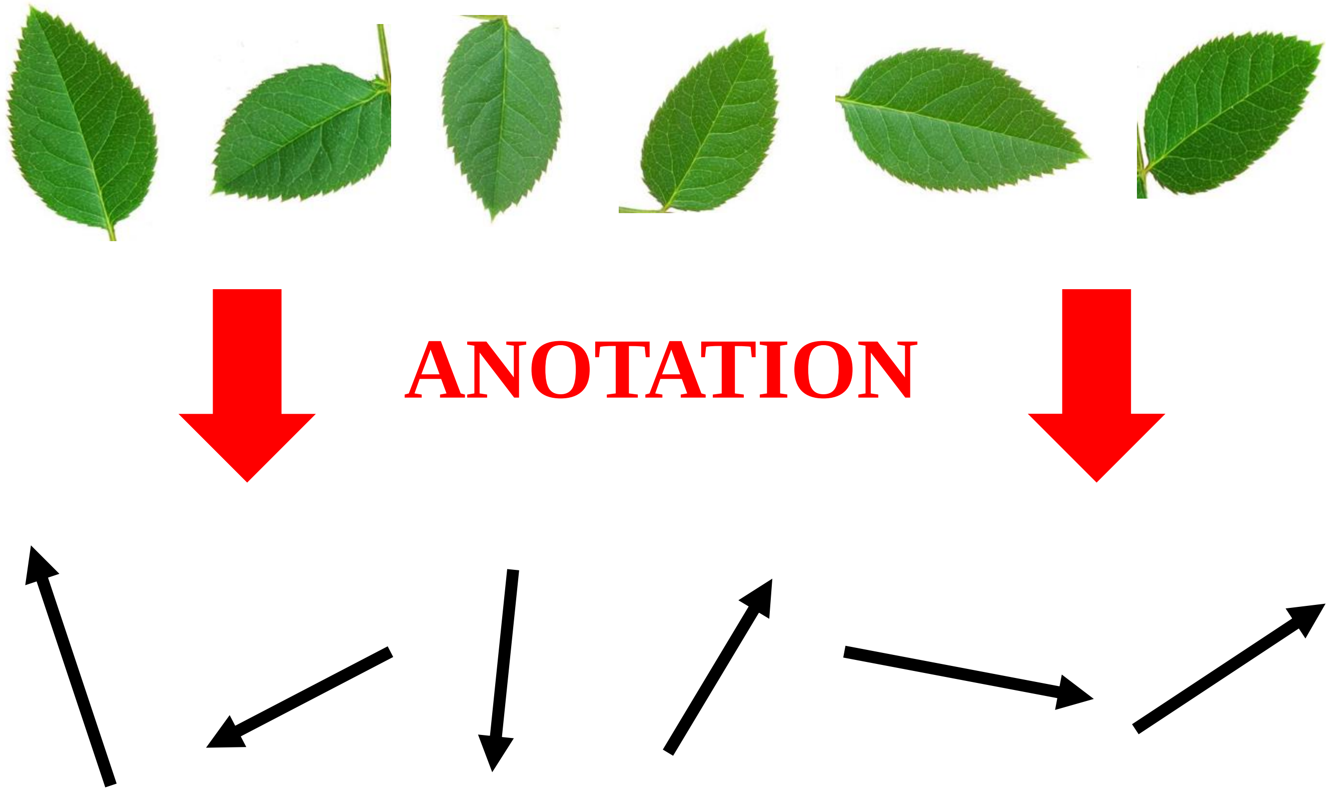
Department of Applied Informatics

Comenius University, Bratislava, Slovakia

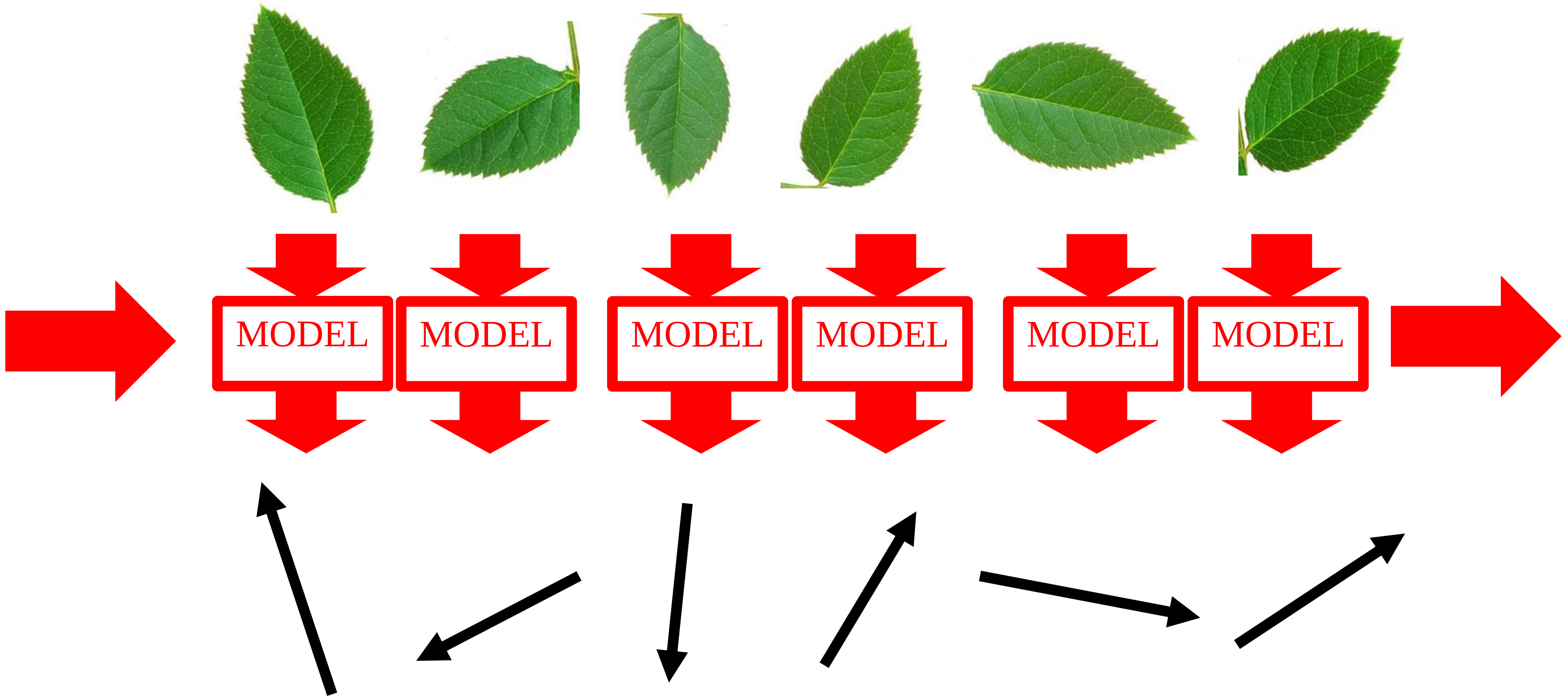
lucny@fmph.uniba.sk

http://dai.fmph.uniba.sk/w/Andrej_Lucny

Dataset

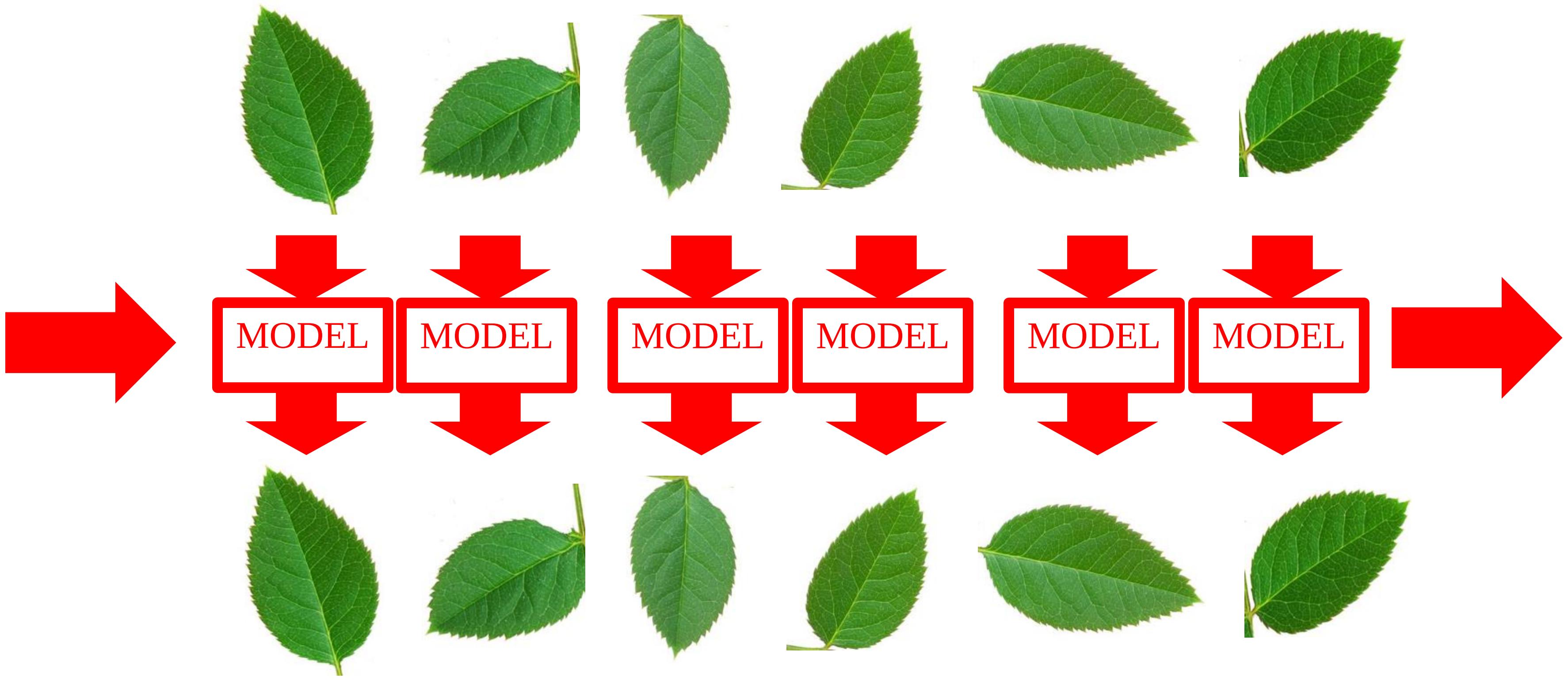


Deep learning



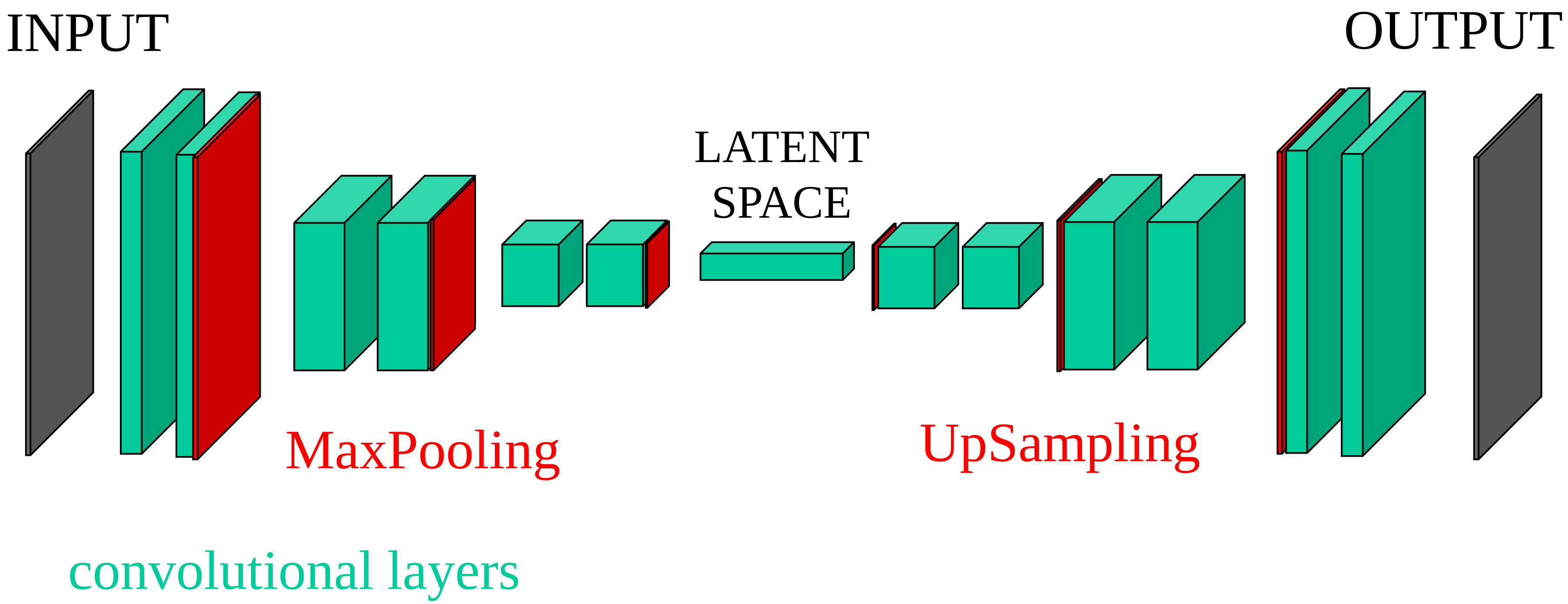
TRAINING in a loop: from the difference between actual and wished outputs, we derive how to modify the model weights to decrease it

Autoencoder



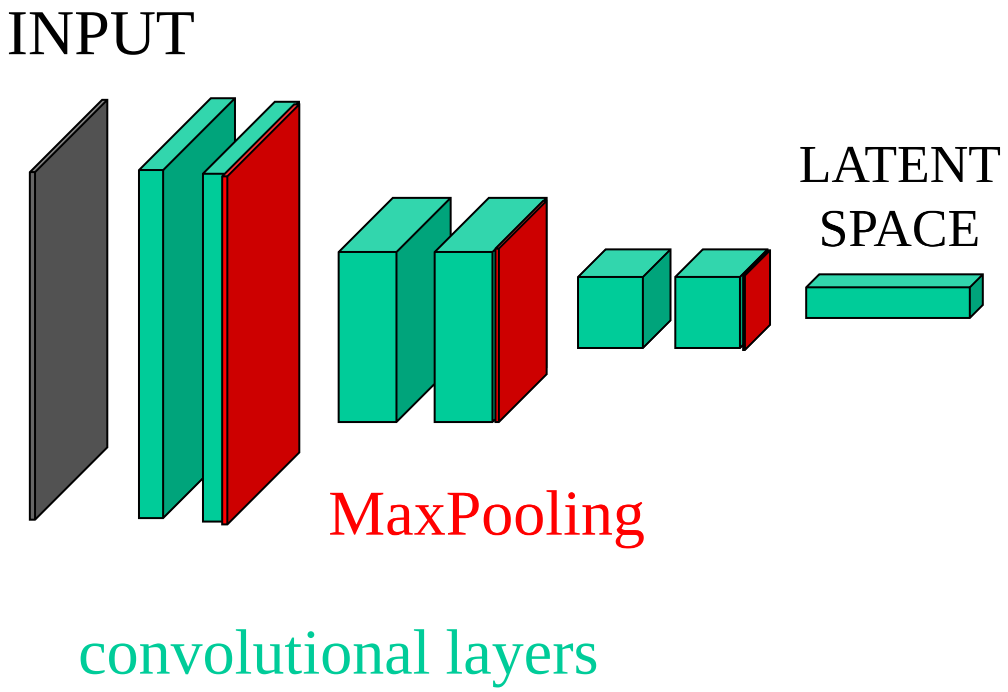
We wish to get on output the same images as on input.
Is it useful for something? Oh, yes, it is.

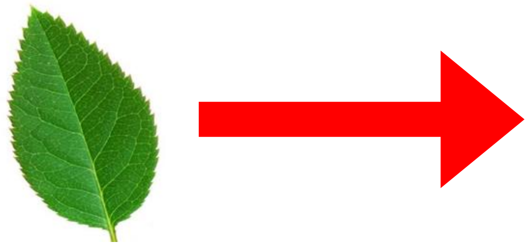
Autoencoder architecture



If we successfully train the autoencoder, the feature vectors in the latent space have to contain the same information images from the dataset. We can split the network into encoder and decoder now.

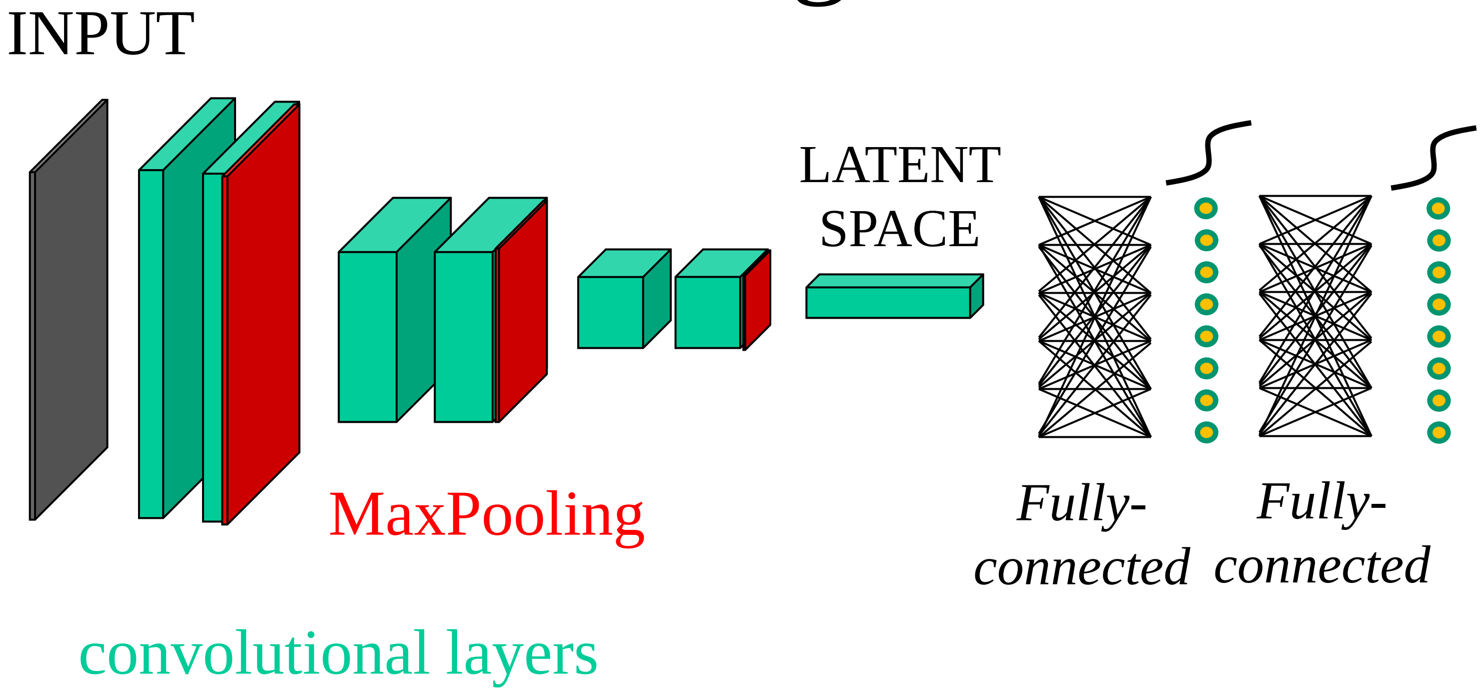
Encoder

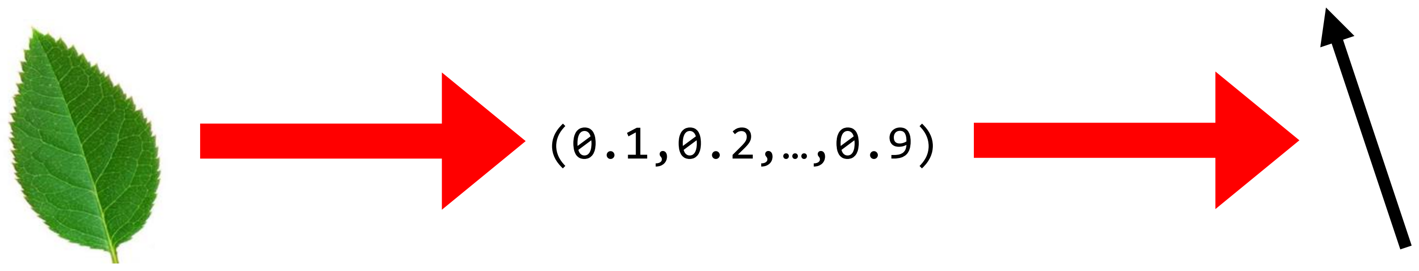




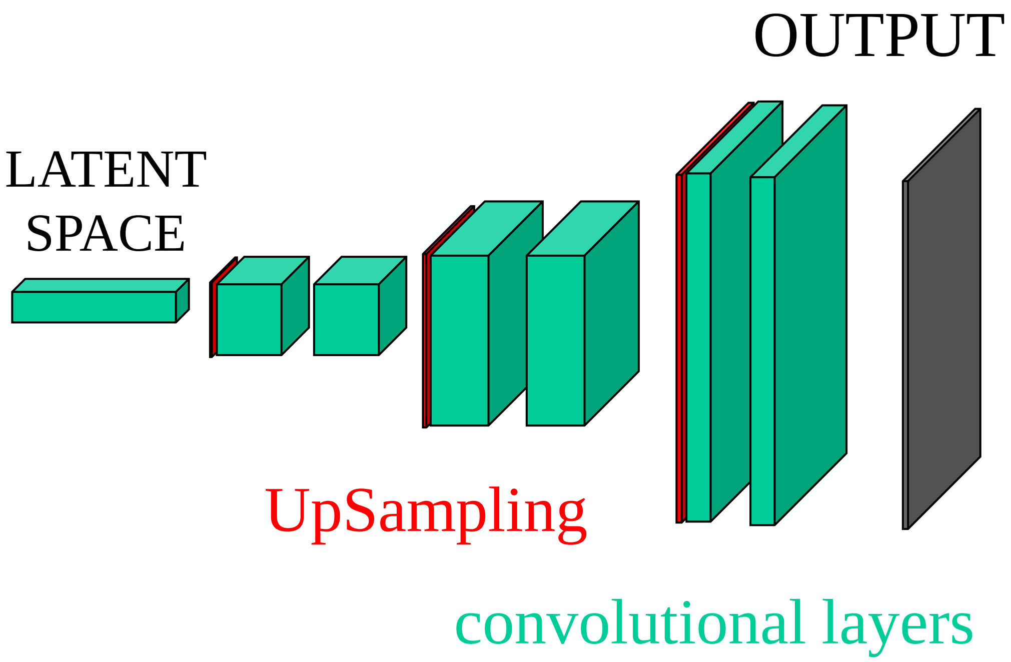
(0.1,0.2,…,0.9)

Regressor

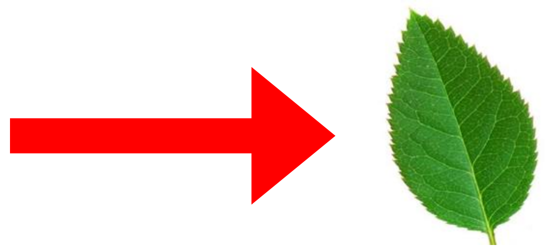

INPUT
LATENT SPACE
MaxPooling
Fully-connected
Fully-connected
convolutional layers


(0.1,0.2,…,0.9)

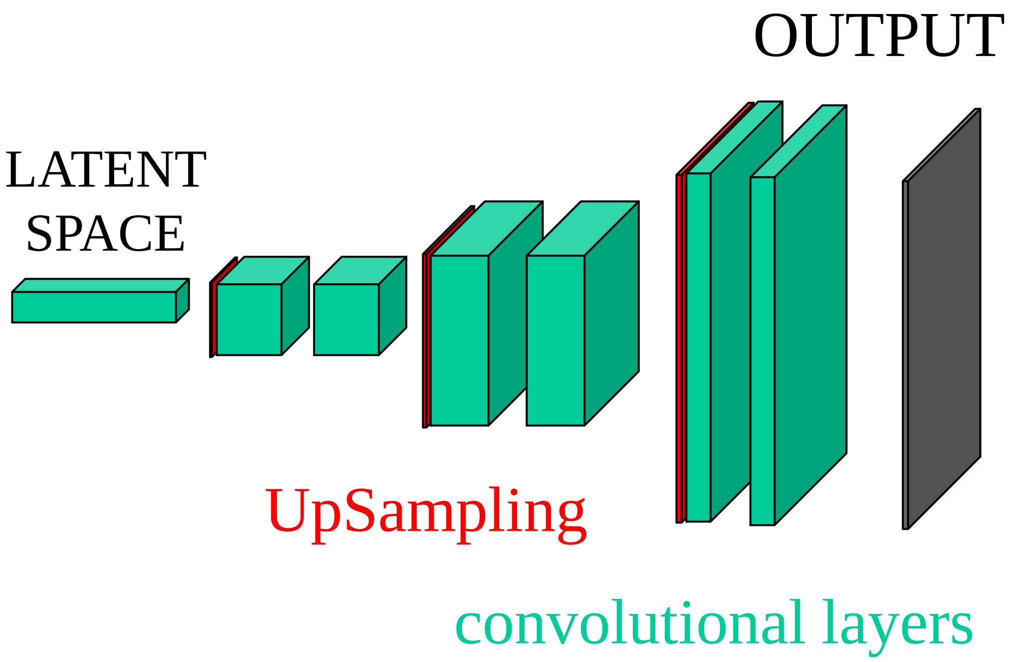
Decoder



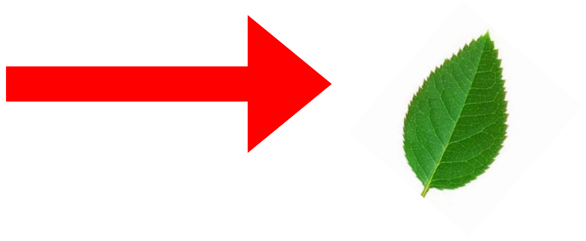
(0.1,0.2,…,0.9)



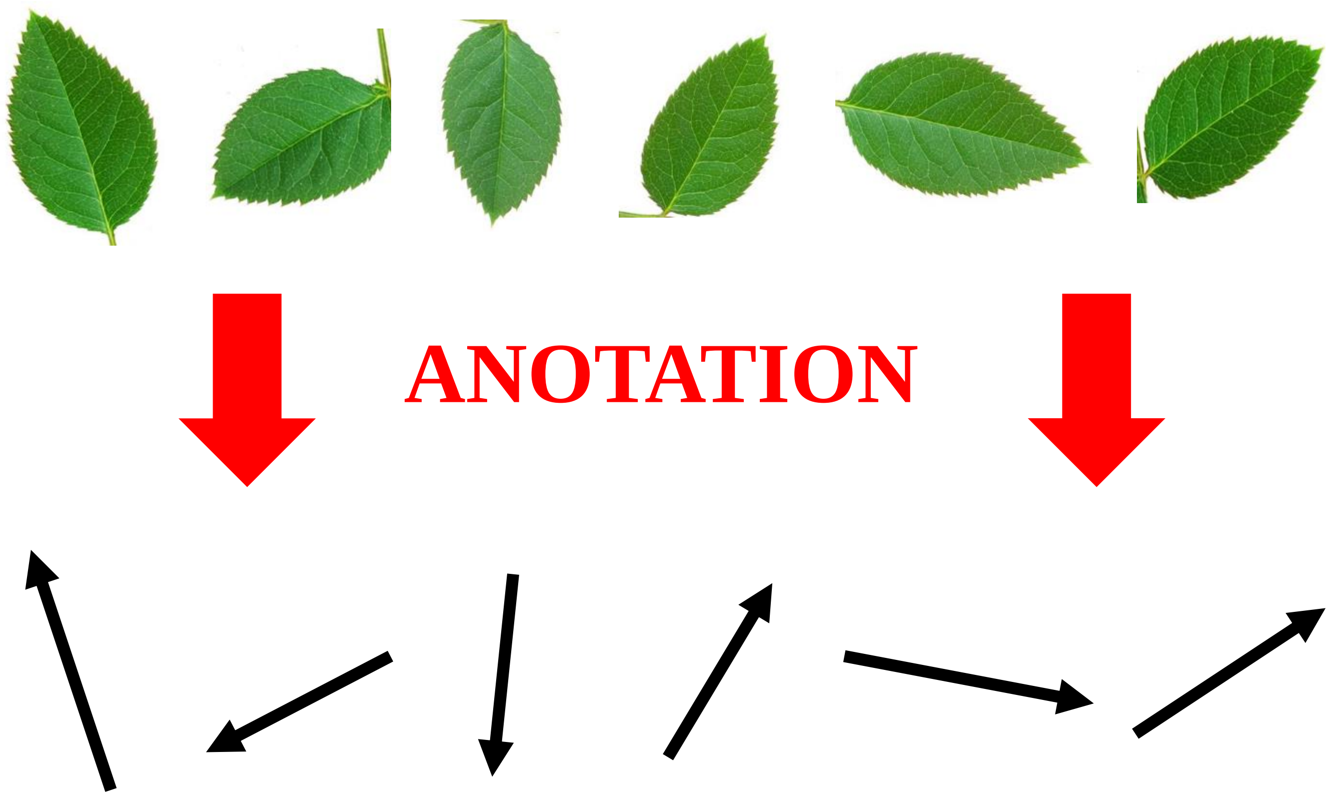
Generator



(0.1,~~0.2~~ 0.5,…,0.9)`

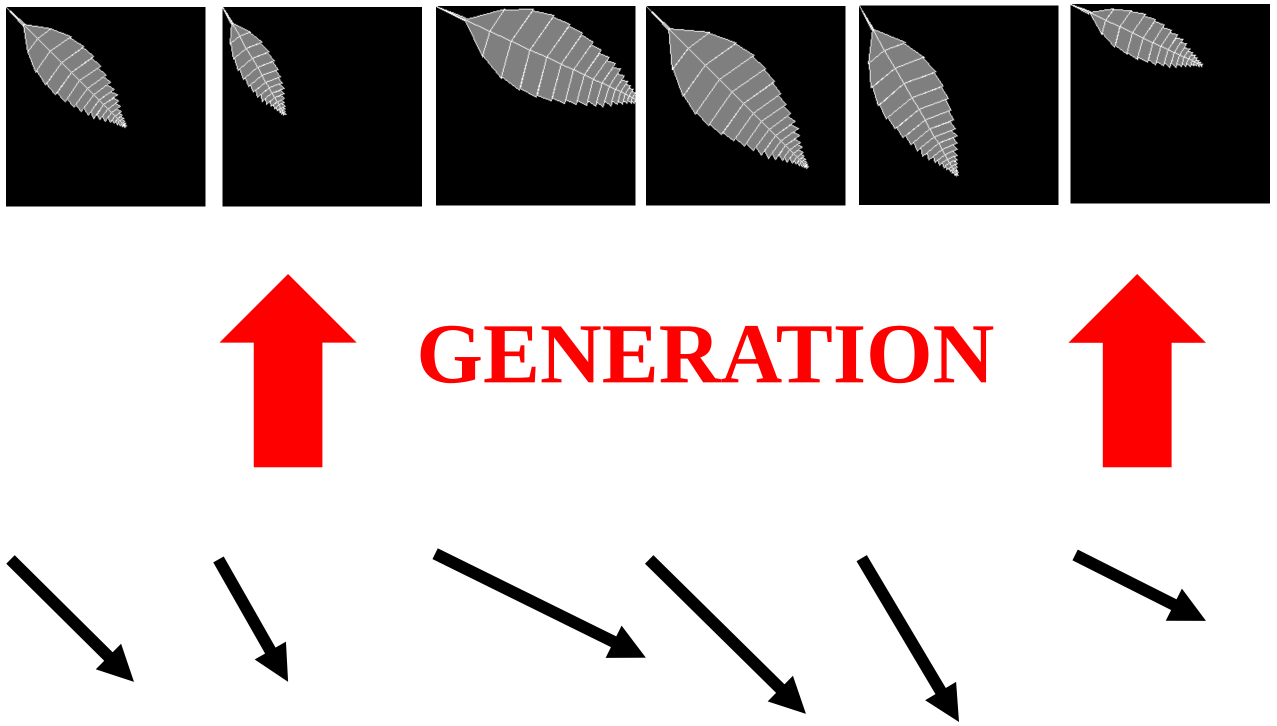


Datasets have many parameters



Annotations do not fully describe the images

Generated datasets have a very exact and low number of parameters



Annotations do fully describe the images

Parametric Lindenmayer systems

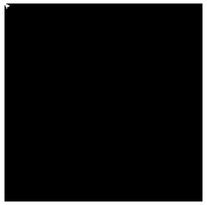
- can generate image datasets from a few parameters

$$
\begin{array}{lll}
\omega_0 & : & [\{A(0,0).\}][\{A(0,1).\}] \\
p1 & : & A(t,d) : d=0 \rightarrow .G(LA,RA). \\
 & & [+B(t)G(LC,RC,t).\}][+B(t)\{.]A(t+1,d) \\
p2 & : & A(t,d) : d=1 \rightarrow .G(LA,RA). \\
 & & [-B(t)G(LC,RC,t).\}][-B(t)\{.]A(t+1,d)| \\
p3 & : & B(t) : t>0 \rightarrow G(LB,RB)B(t-1) \\
p4 & : & G(s,r) \rightarrow G(s*r,r) \\
p5 & : & G(s,r,t) : t>1 \rightarrow G(s*r,r,t-1)
\end{array}
$$

rose leaves *[Prusinkiewicz, Lindenmayer 1990]*

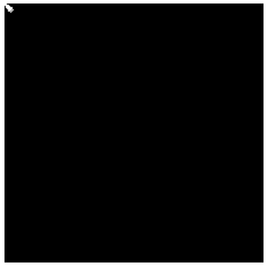
```
[{A(0,0).}][{A(0,1).}]
```

```
[{.G(5,1.15).[+B(0)G(3,1.19,0).}][+B(0){.
]A(1,0).}][{.G(5,1.15).[-B(0)G(3,1.19,0)
.}][-B(0){.]A(1,1).}]
```



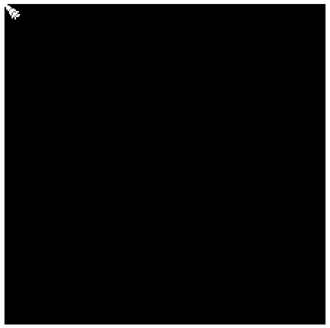
iteration 1

```
[{.G(5.75,1.15).[+B(0)G(3,1.19,0).}][+B(0
){.].G(5,1.15).[+B(1)G(3,1.19,1).}][+B(1)
{.]A(2,0).}][{.G(5.75,1.15).[-B(0)G(3,
1.19,0).}][-B(0){.].G(5,1.15).[-B(1)
G(3,1.19,1).}][-B(1){.]A(2,1).}]
```



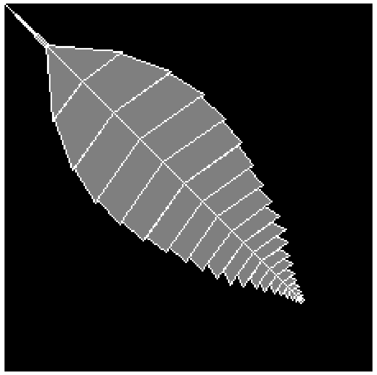
iteration 2

```
[{.G(6.6125,1.15).[+B(0)G(3,1.19,0).}][+B
(0){.].G(5.75,1.15).[+G(1.3,1.25)B(0)G(3,
1.19,1).}][+G(1.3,1.25)B(0){.].G(5,1.15).
[+B(2)G(3,1.19,2).}][+B(2){.]A(3,0).}][{.
G(6.6125,1.15).[-B(0)G(3,1.19,0).}][-
B(0){.].G(5.75,1.15).[-G(1.3,1.25)B(0)G …
```



iteration 3

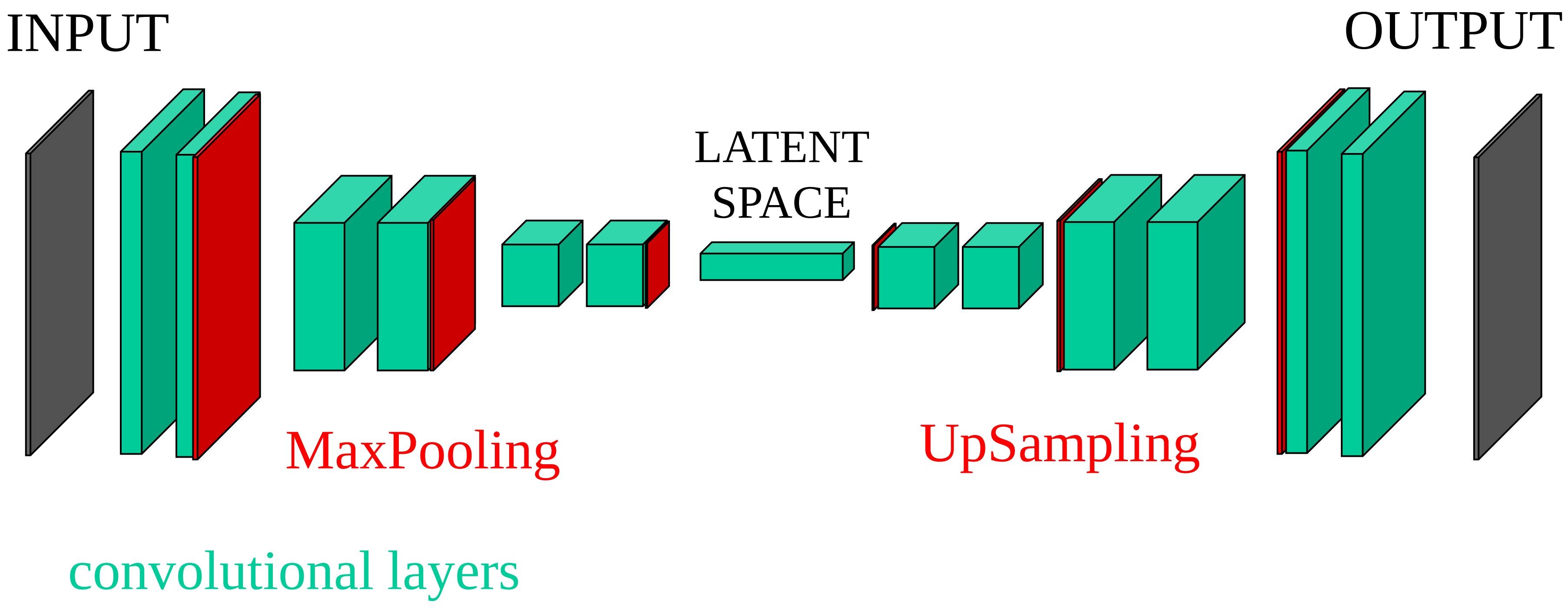
```
[+G(3.173828125,1.25)G(2.5390625,1.25)G(2
.03125,1.25)G(1.625,1.25)G(1.3,1.25)B(8)G
(7.15906097969998,1.19,8).}][+G(3.173
828125,1.25)G(2.5390625,1.25)G(2.03125,1.
25)G(1.625,1.25)G(1.3,1.25)B(8){.].G(8.74
503124999998,1.15).[+G(2.5390625,1.25
)G(2.03125,1.25)G(1.625,1.25)G(1.3,1.25)B
(10)G(6.01601762999999,1.19,10).}][+G(2.
5390625,1.25)G(2.03125,1.25)G(1.625,1.
25)G(1.3,1.25)B(10){.].G(7.6043749999999
9,1.15).[+G(2.03125,1.25)G(1.625,1.25)G(1
.3,1.25)B(12)G(5.05547699999999,1.19,
12).}][+G(2.03125,1.25)G(1.625,1.25)G(1.3
,1.25)B(12){.].G(6.6125,1.15).[+G(1.625,1
.25)G(1.3,1.25)B(14)G(4.2482999999999
95,1.19,14).}][+G(1.625,1.25)G(1.3,1.25)B
(14){.].G(5.75,1.15).[+G(1.3,1.25)B(16)G(
3.57,1.19,16).}][+G(1.3,1.25)B(16){.]. …
```

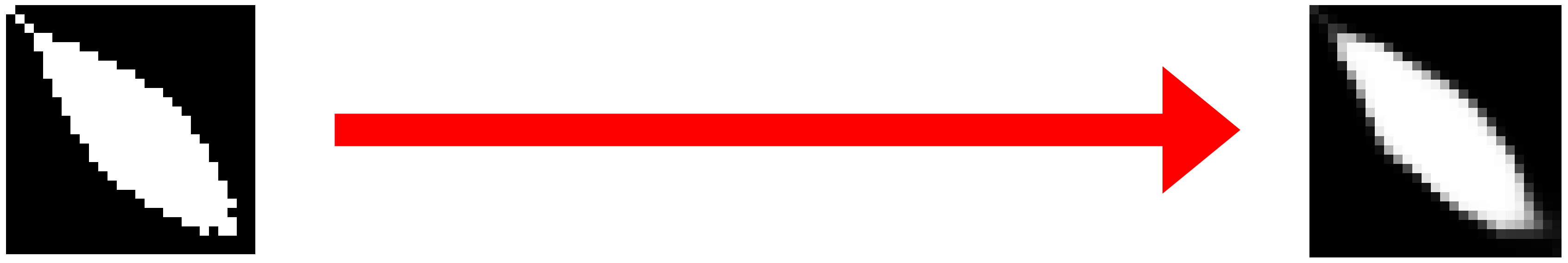

iteration 19

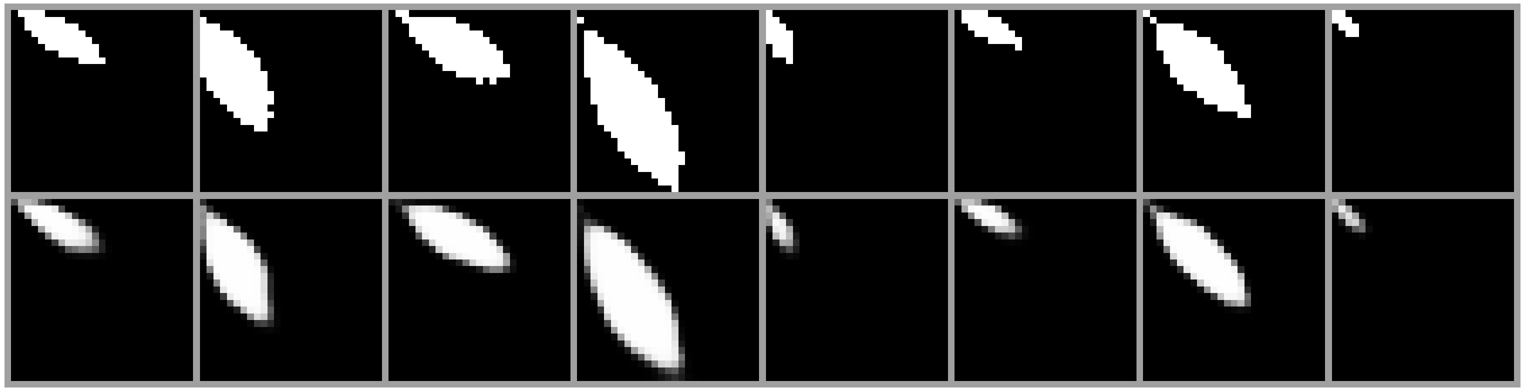
We have investigated:

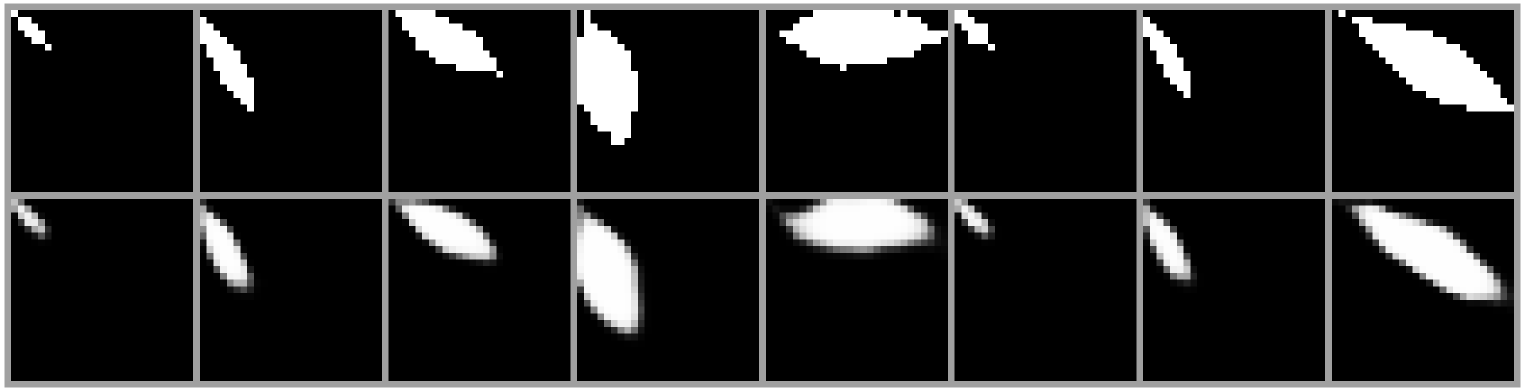
1. Can we find the parameters of the Lindenmayer system somewhere inside the latent space of the neural network that is processing a dataset produced by the Lindenmayer system?
2. Can we create a neural network that generates the same images as the Lindenmayer system?
3. And could a neural network make the images from the parameters of the Lindenmayer system?

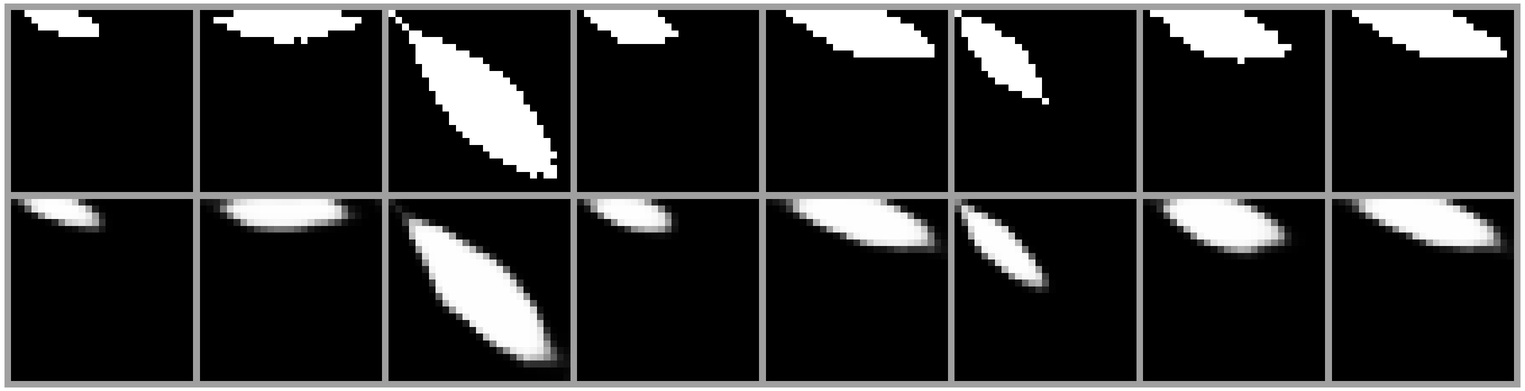
We have trained autoencoder





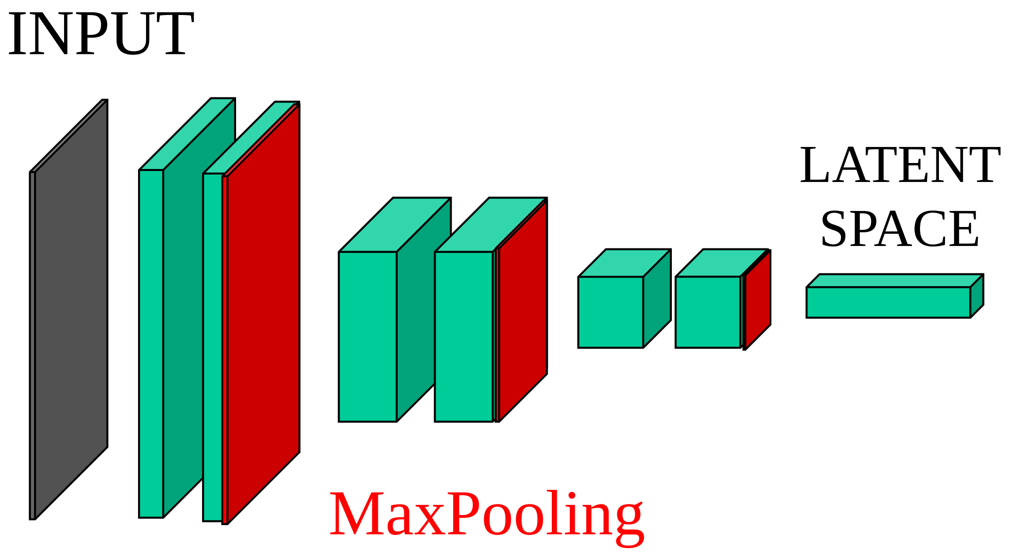


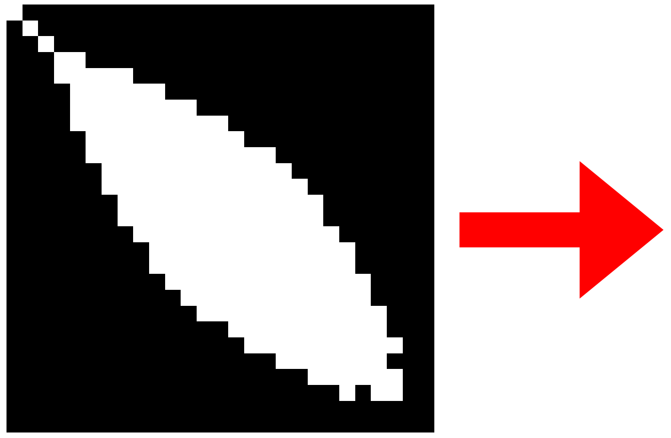




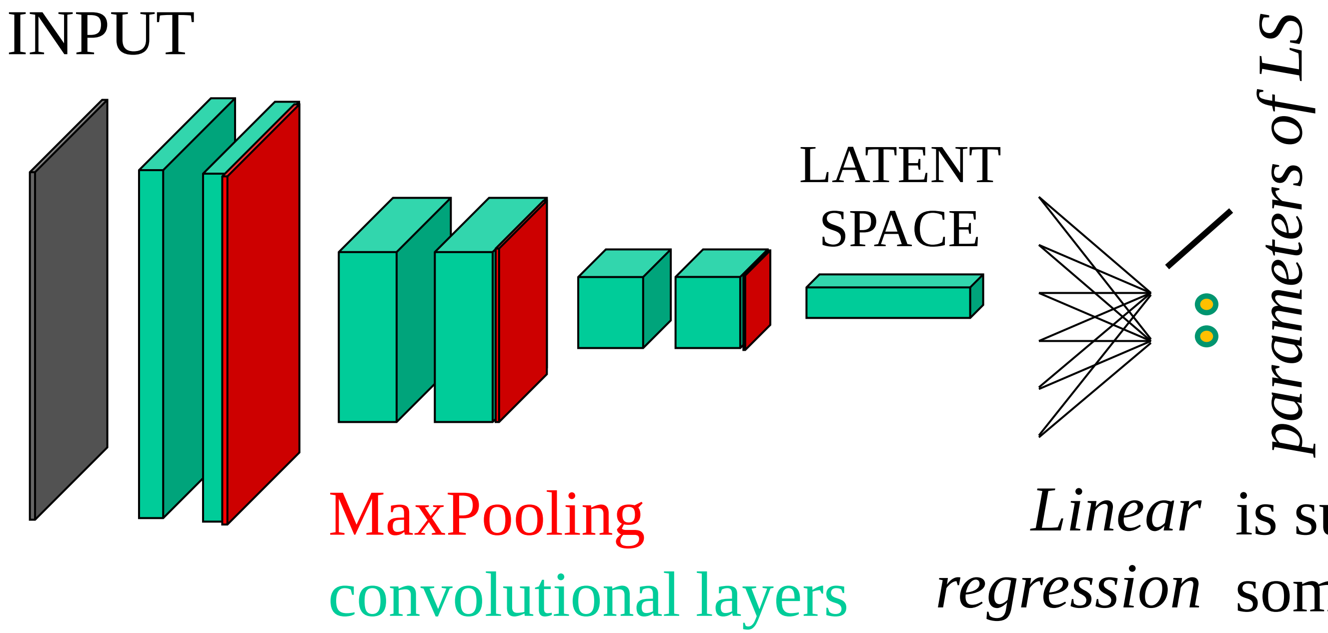
test accuracy 98,6%

Can we find the parameters of the Lindenmayer system somewhere inside the latent space of the neural network that is processing a dataset produced by the Lindenmayer system?



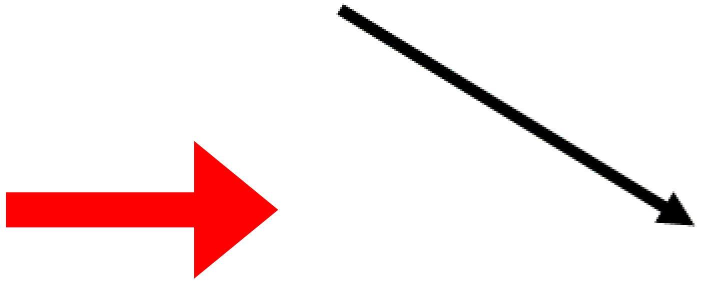


Can we find the parameters of the Lindenmayer system somewhere inside the latent space of the neural network that is processing a dataset produced by the Lindenmayer system?

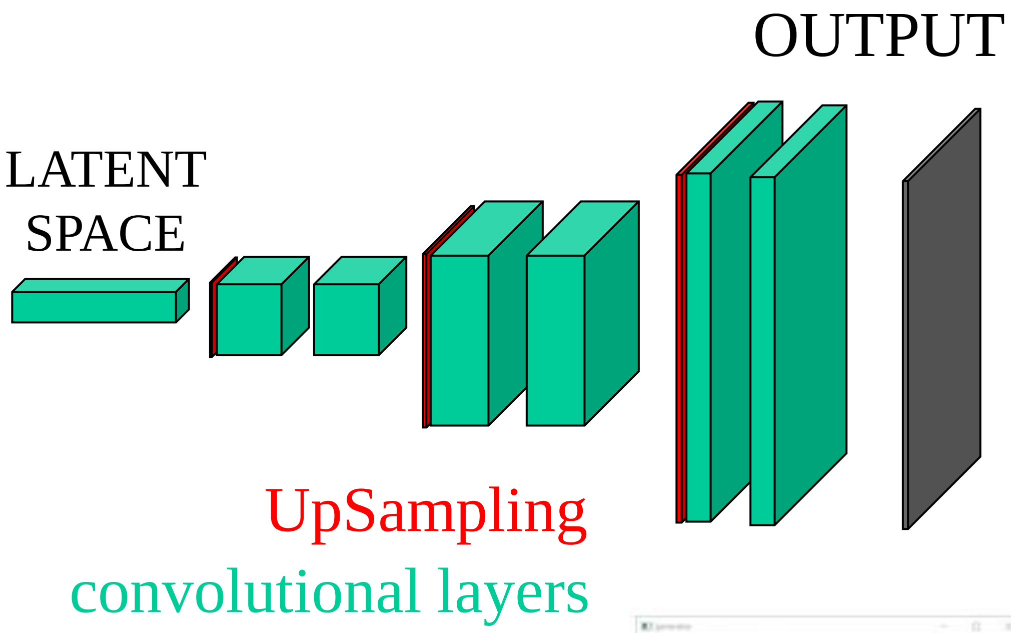


is suffient to reveal some paramaters of LS

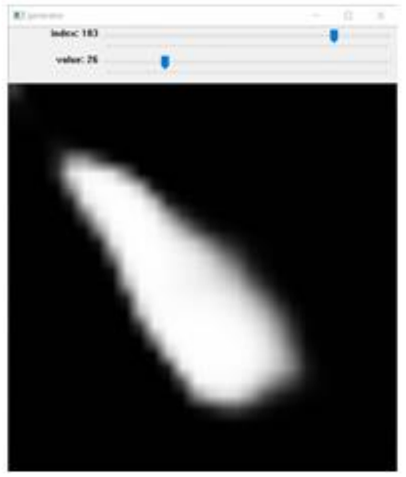
(0.1,0.2,…,0.9)



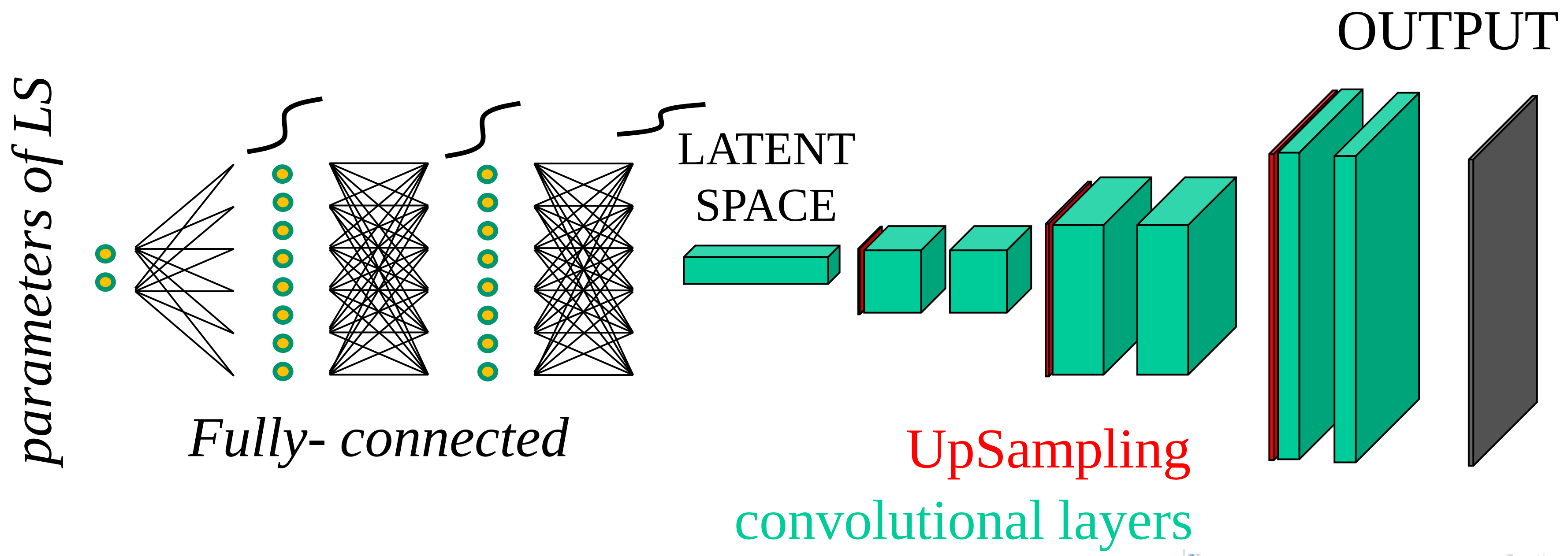
Can we create a neural network that generates the same images as the Lindenmayer system?



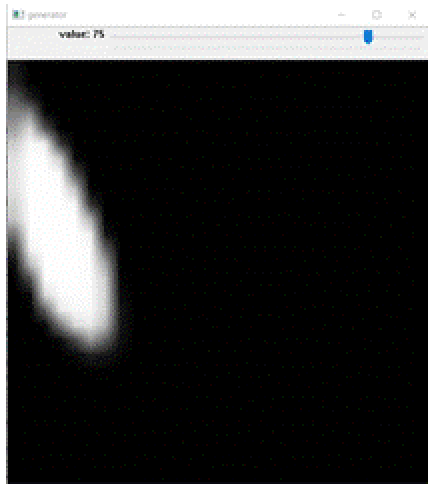
0.5
(0.1,~~0.2~~,…,0.9)`



And could a neural network make the images from the parameters of the Lindenmayer system?



(0.1,0.5,…,0.9)`



Implementation details

TensorFlow

Keras

OpenCV

python™

https://github.com/andylucny/
On-Lindenmayer-Systems-and-Autoencoders.git

Further development

- Autoencoder architecture for higher image resolution
- Not only binary images
- Reduction of the latent space dimension
- Further investigation and better visualization of the latent space
- A simpler network for generation from parameters

Thank you!

On Lindenmayer Systems and Autoencoders

Andrej Lúčny

Department of Applied Informatics

Comenius University, Bratislava, Slovakia

lucny@fmph.uniba.sk

http://dai.fmph.uniba.sk/w/Andrej_Lucny

https://github.com/andylucny/On-Lindenmayer-Systems-and-Autoencoders.git