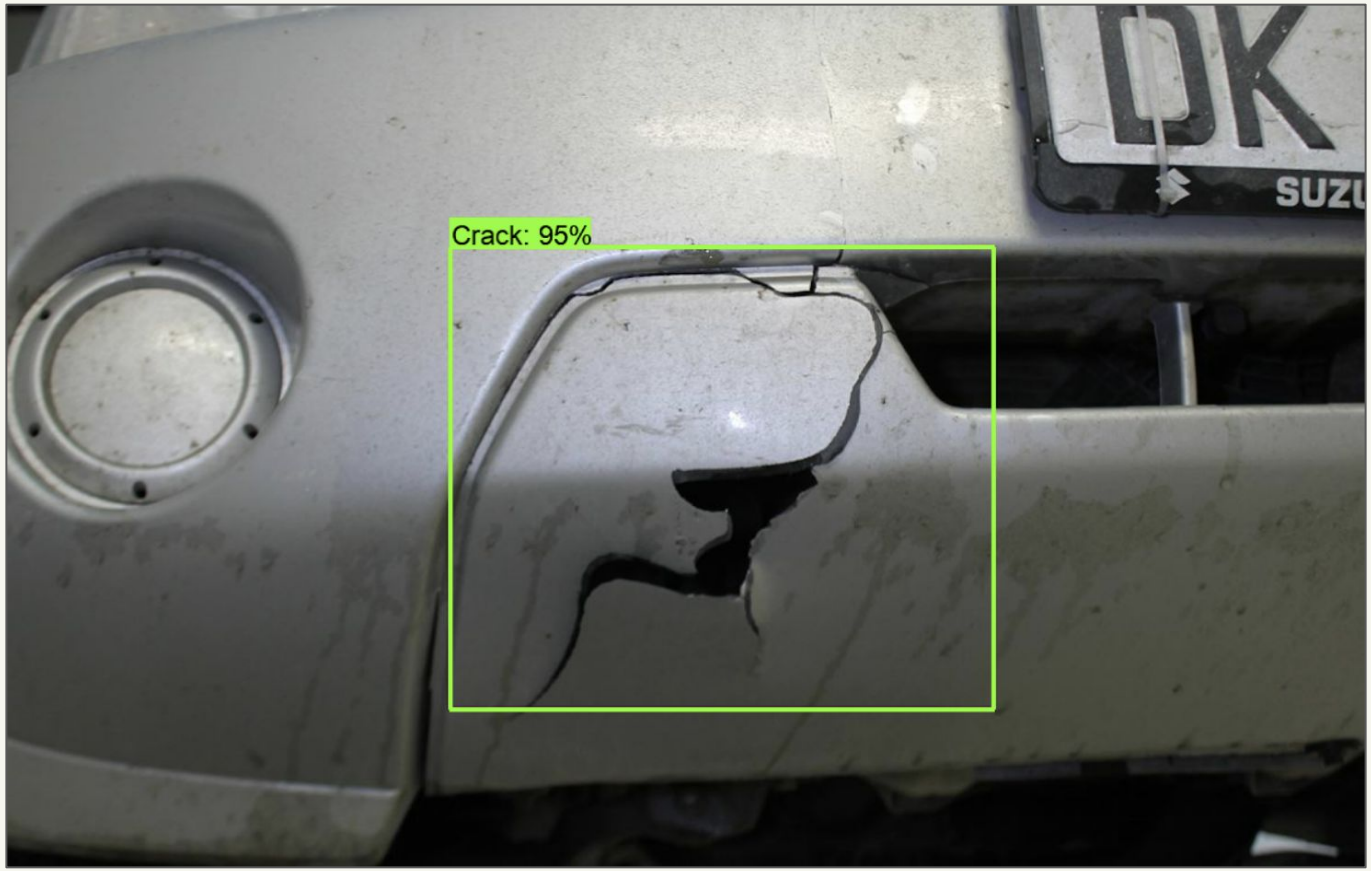
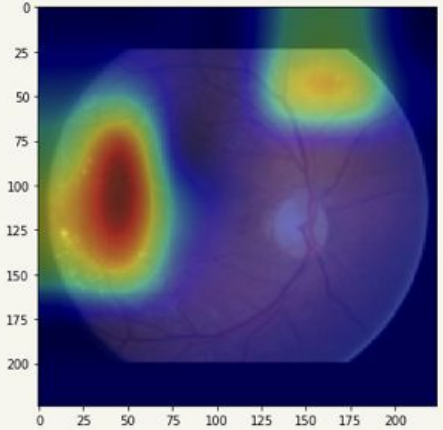
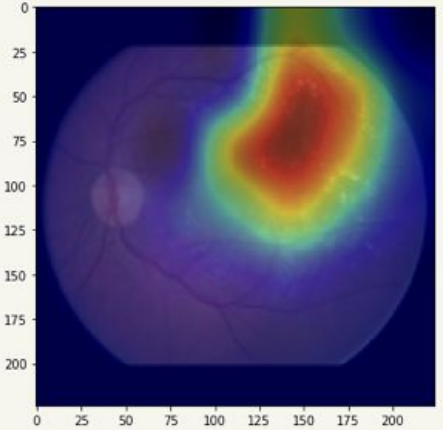
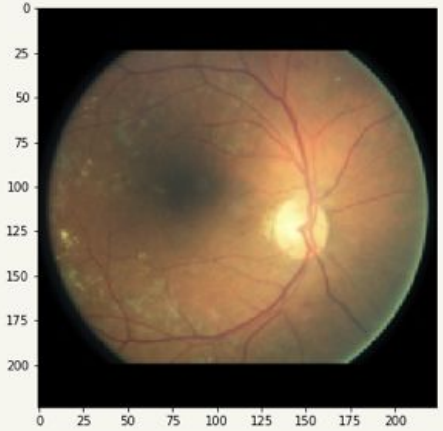
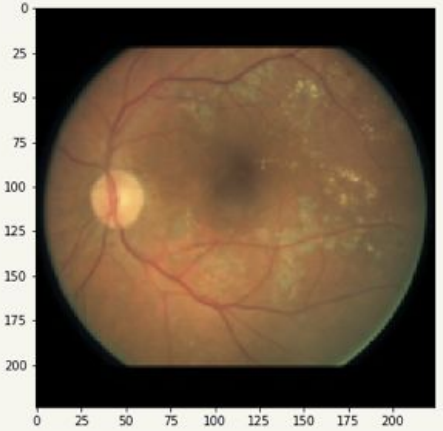
Image recognition in Python

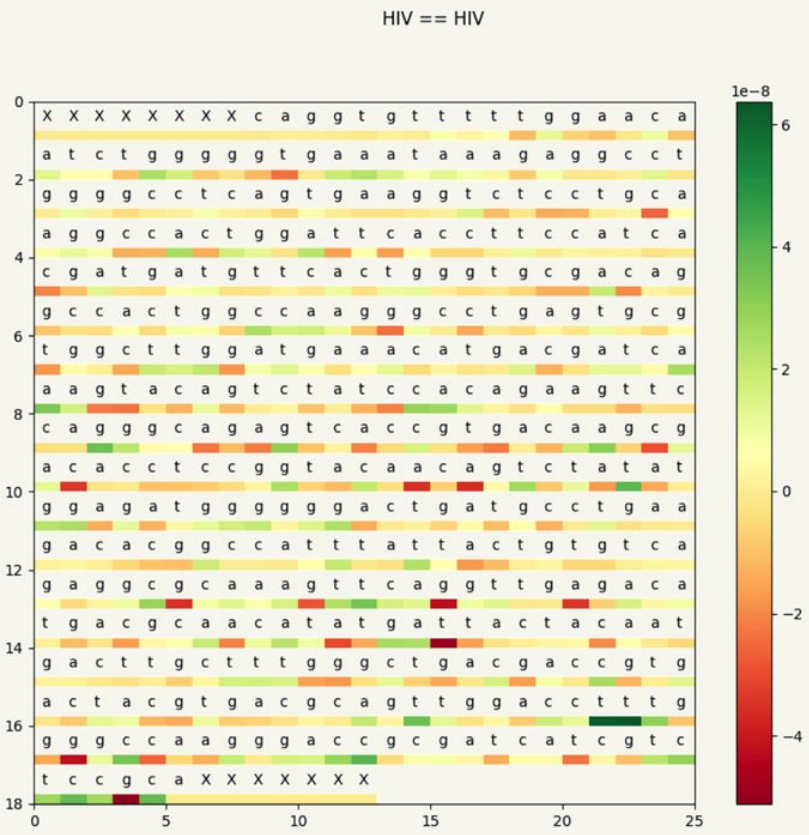

Crack: 95%
DK


0
25
50
75
100
125
150
175
200
0 25 50 75 100 125 150 175 200


0
25
50
75
100
125
150
175
200
0 25 50 75 100 125 150 175 200


0
25
50
75
100
125
150
175
200
0 25 50 75 100 125 150 175 200


0
25
50
75
100
125
150
175
200
0 25 50 75 100 125 150 175 200


HIV == HIV
1e−8
6
4
2
0
−2
−4
0
5
10
15
20
25
X X X X X X X c a g g t g t t t t g g a a c a
a t c t g g g g g t g a a a t a a a g a g g c c t
g g g g c c t c a g t g a a g g t c t c c t g c a
a g g c c a c t g g a t t c a c c t t c c a t c a
c g a t g a t g t t c a c t g g g t g c g a c a g
g c c a c t g g c c c a a g g g c c t g a g t g c g
t g g c t t g g a t g a a a c a t g a c g a t c a
a a g t a c a g t c t a t c c a c a g a a g t t c
c a g g g c a g a g a g t c a c c g t g a c a a g c g
a c a c c t c c g g t a c a a c a g t c t a t a t
g g a g a t g g g g g g g a c t g a t g c c t g a a
g a c a c g g c c c a t t t a t t a c t g t g t c a
g a g g c g c a a a a g t t c a g g t t g a g a c a
t g a c g c a a c a t a t g a t t a c t a c a a a t
g a c t t g c t t t g g g c t g a c g a c c g t g
a c t a c g t g a c g c a g t t g g a c c t t t g
g g g c c a a g g g g a c c g c g a t c a t c g t c
t c c g c a X X X X X X X

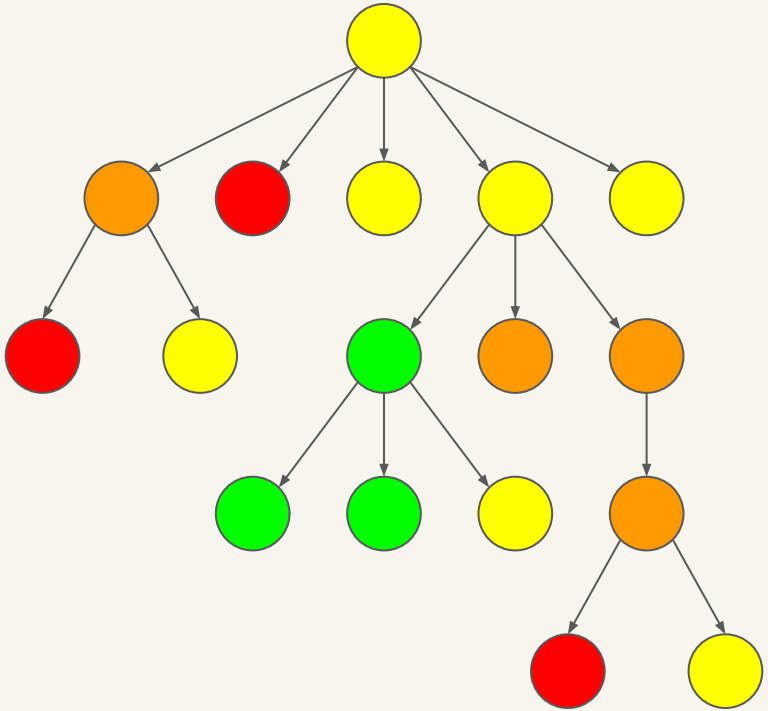
Thank you

I really appreciate the help :)

Amazing!!

What the !*%$ is going on?!?

Thanks for nothing. What a horrible user experience



Machine learning

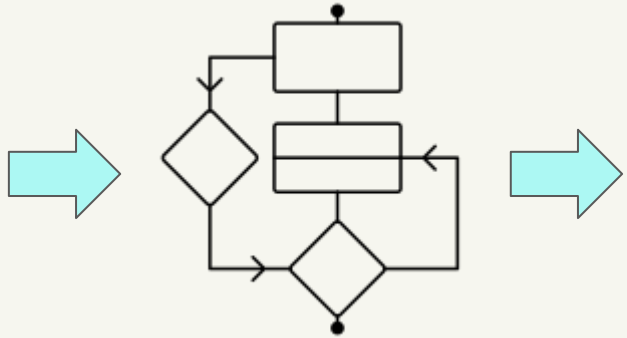
EPIGRAM MEDTECH

```
procedure bubbleSort( A : list of sortable items )
    n = length(A)
    repeat
        swapped = false
        for i = 1 to n-1 inclusive do
            /* if this pair is out of order */
            if A[i-1] > A[i] then
                /* swap them and remember something changed */
                swap( A[i-1], A[i] )
                swapped = true
            end if
        end for
    until not swapped
end procedure
```

```
procedure bubbleSort( A : list of sortable items )
    n = length(A)
    repeat
        swapped = false
        for i = 1 to n-1 inclusive do
            /* if this pair is out of order */
            if A[i-1] > A[i] then
                /* swap them and remember something changed */
                swap( A[i-1], A[i] )
                swapped = true
            end if
        end for
    until not swapped
end procedure
```

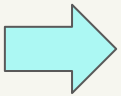
```
def bubblesort(l: List) -> List:
    while True:
        swapped = False

        for i in range(len(l) - 1):
            if l[i] > l[i + 1]:
                tmp = l[i]
                l[i] = l[i + 1]
                l[i + 1] = tmp
                swapped = True

        if not swapped:
            break

    return l
```

Inputs:
[3, 5, 1]
[2]
[2, 3, 4, 5]
[]
[2, 1]

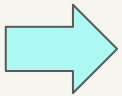


Outputs:
[1, 3, 5]
[2]
[2, 3, 4, 5]
[]
[1, 2]

Inputs:
[3, 5, 1]
[2]
[2, 3, 4, 5]
[]
[2, 1]



?



Outputs:
[1, 3, 5]
[2]
[2, 3, 4, 5]
[]
[1, 2]

FINN Mulighetenes marked

Varslinger | Ny annonse | Meldinger | Min FINN

Eiendom / Bolig til salgs

527 treff i **202** annonser

Lagre søk | Kart | Sortér på Publisert

sofienberg

Publisert

Nye i dag (68)

Område, by eller sted

Søk etter adresse eller sted

Område

- Akershus (0)
- Aust-Agder (0)
- Buskerud (0)
- Finnmark (0)
- Hedmark (0)
- Hordaland (0)
- Møre og Romsdal (0)
- Nordland (0)

Ukens bolig | Visning i dag

Markveien 1 A, Oslo, Grünerløkka - Sofienberg

ROLIG PÅ ØVRE GRÜNERLØKKA: Lys og lekker 3-roms loftsleilighet med privat solrik takterrasse og felles takterrasse. -Idyllisk bakgård -Hems -

78 m² **6 200 000,-**

Andel • Leilighet • 2 soverom
Schala & Partners Grünerløkka
Fellesgjeld: 28 064,- • Fellesutg.: 5 226,-

SCHALA & PARTNERS

Sverdrups gate 10A, Oslo, Grünerløkka - Sofienberg

GRÜNERLØKKA - Unik 2-roms selveier over tre plan i sjarmerende omgjort stall. Terrasse, romslig hems og herlig town-

42 m² **3 400 000,-**

DNB Eiendom AS
Eier (Selveier) • Leilighet • 1 soverom
Fellesgjeld: 65 000,- • Fellesutg.: 2 215,-

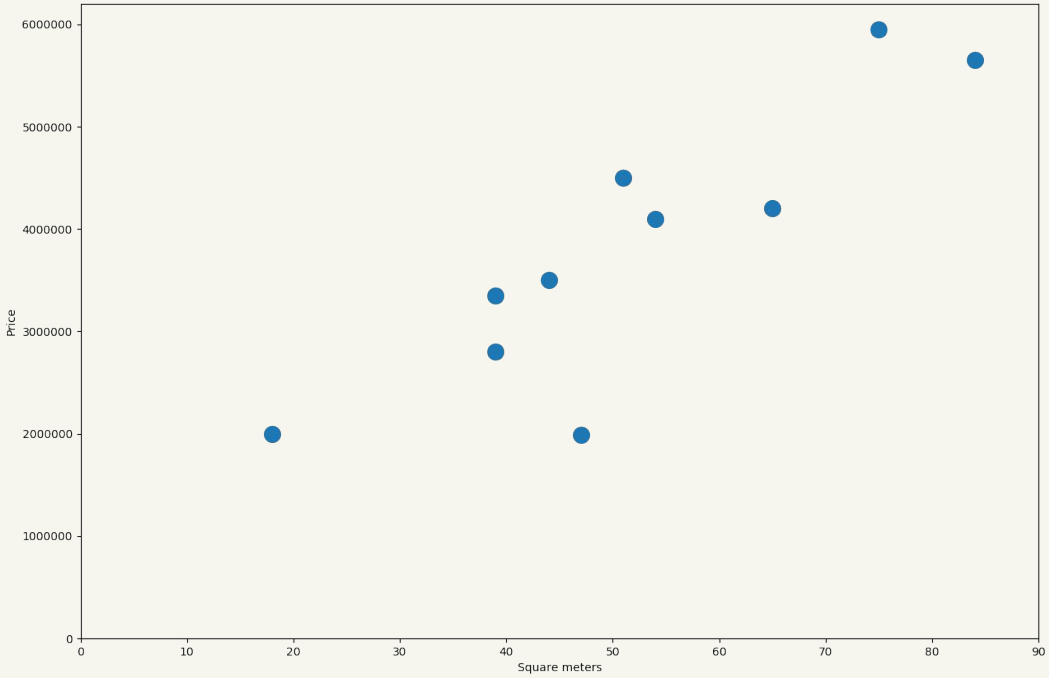
DNB

Visning i dag

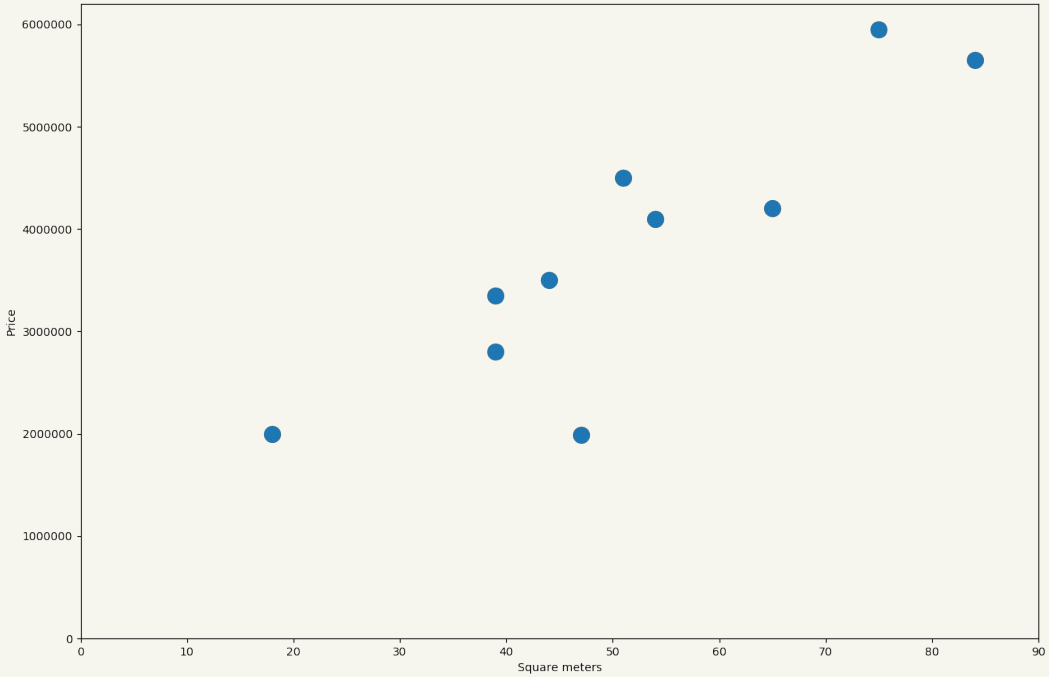
Lørenveien 57-63, leil. B2-35, Oslo

Fastpris: Lørenporten - Ny 2R m/ca. 15 kvm vest-vendt terrasse. 31.12.2020 tas det sikte på over-takelse.

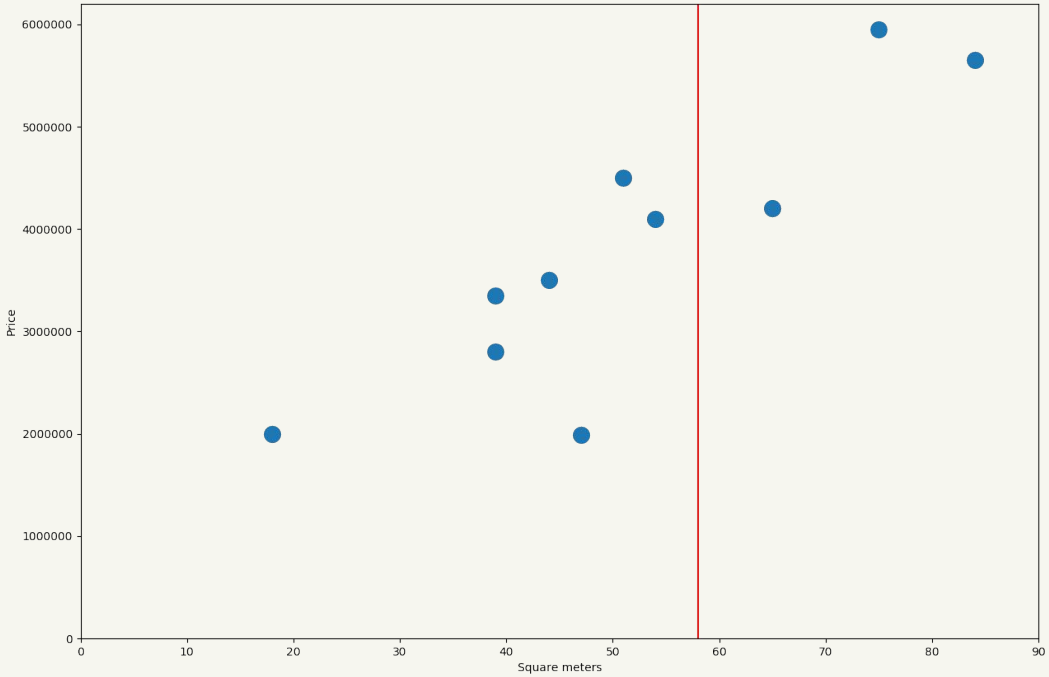
m^2	Price
51	4 500 000
84	5 650 000
44	3 500 000
47	1 990 000
75	5 950 000
39	3 350 000
18	2 000 000
65	4 200 000
54	4 100 000
39	2 800 000


Price
Square meters
6000000
5000000
4000000
3000000
2000000
1000000
0
0
10
20
30
40
50
60
70
80
90

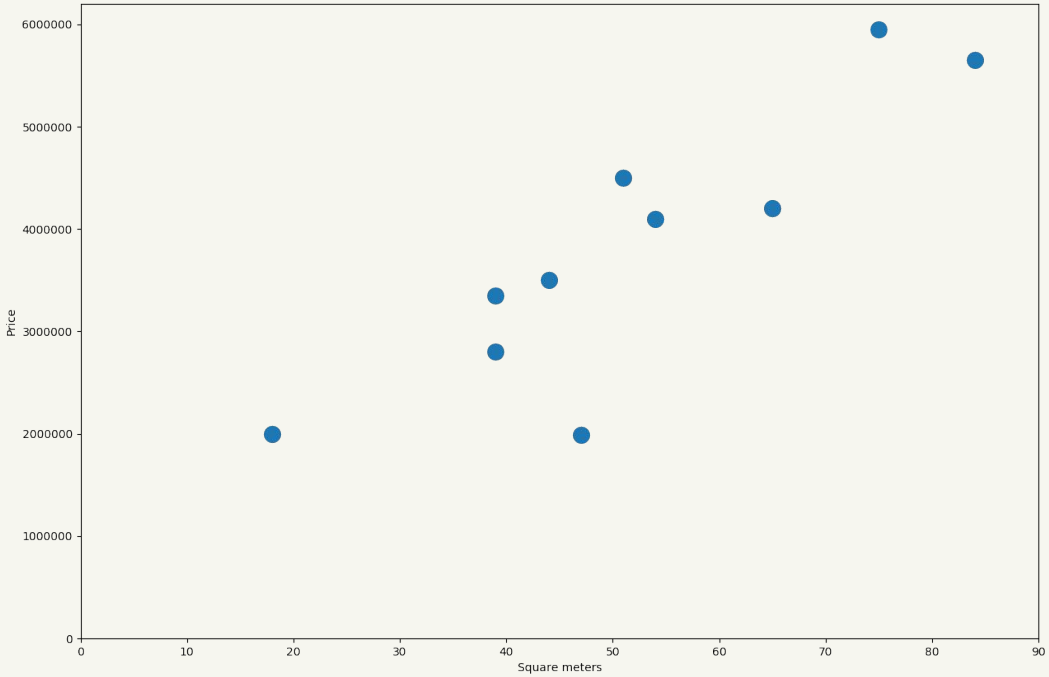
price = f(m^2)


6000000
5000000
4000000
3000000
2000000
1000000
0
Price
0
10
20
30
40
50
60
70
80
90
Square meters

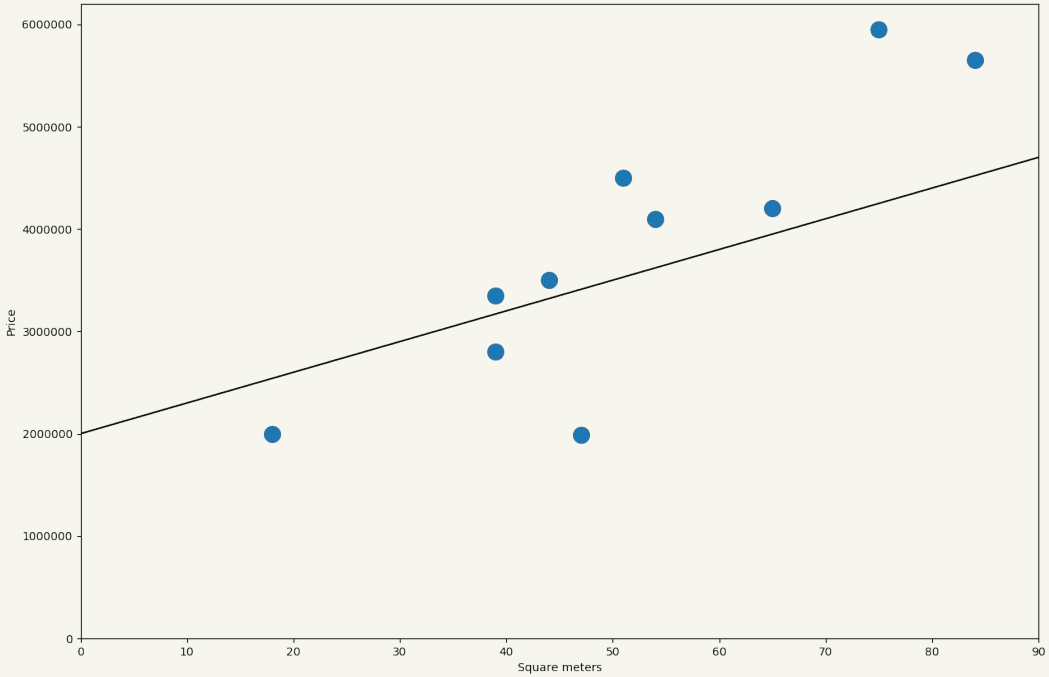
price = f(58)


6000000
5000000
4000000
3000000
2000000
1000000
0
Price
0
10
20
30
40
50
60
70
80
90
Square meters

price = a(m^2)+b


6000000
5000000
4000000
3000000
2000000
1000000
0
Price
0
10
20
30
40
50
60
70
80
90
Square meters

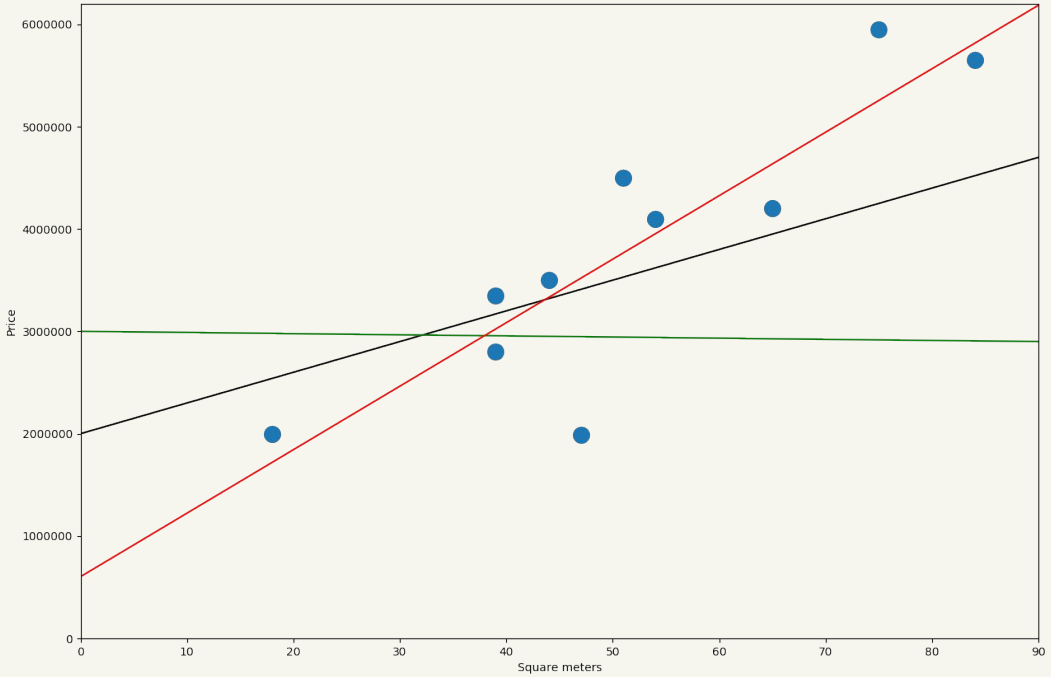
price = 30000(m^2)+2000000



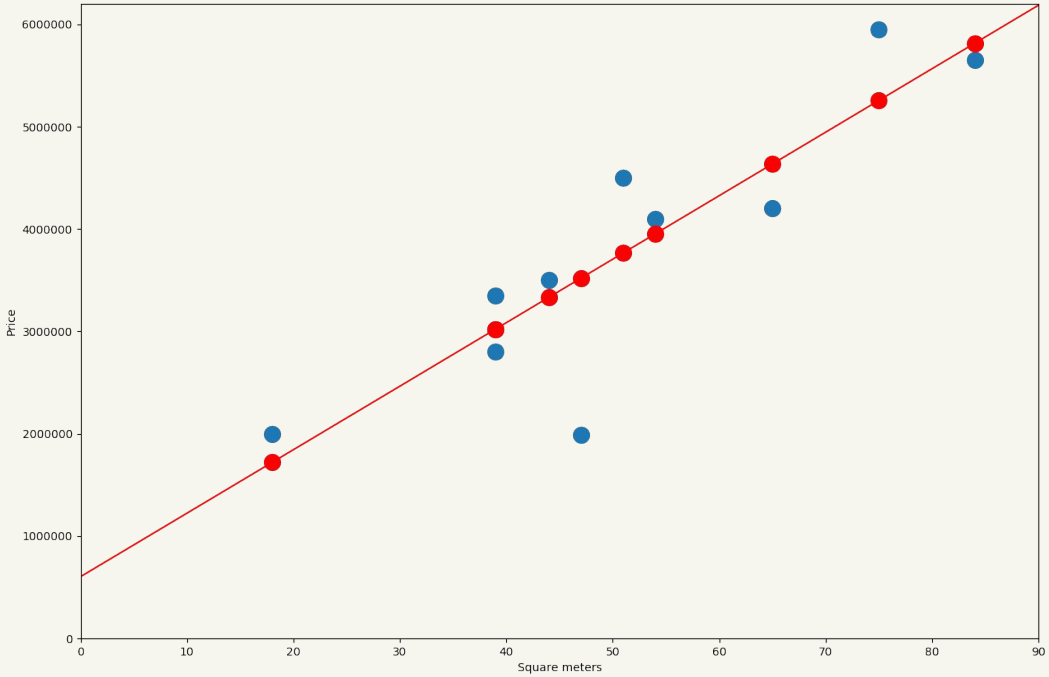
price = 10000(m^2)+3000000

price = 30000(m^2)+2000000

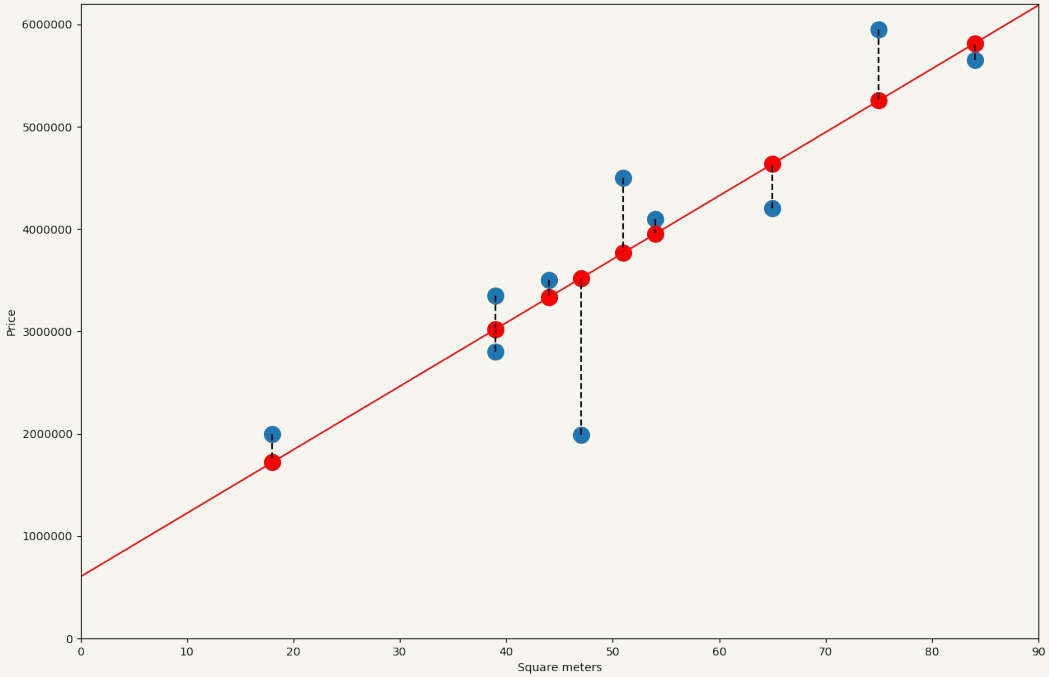
price = 62044(m^2)+602535



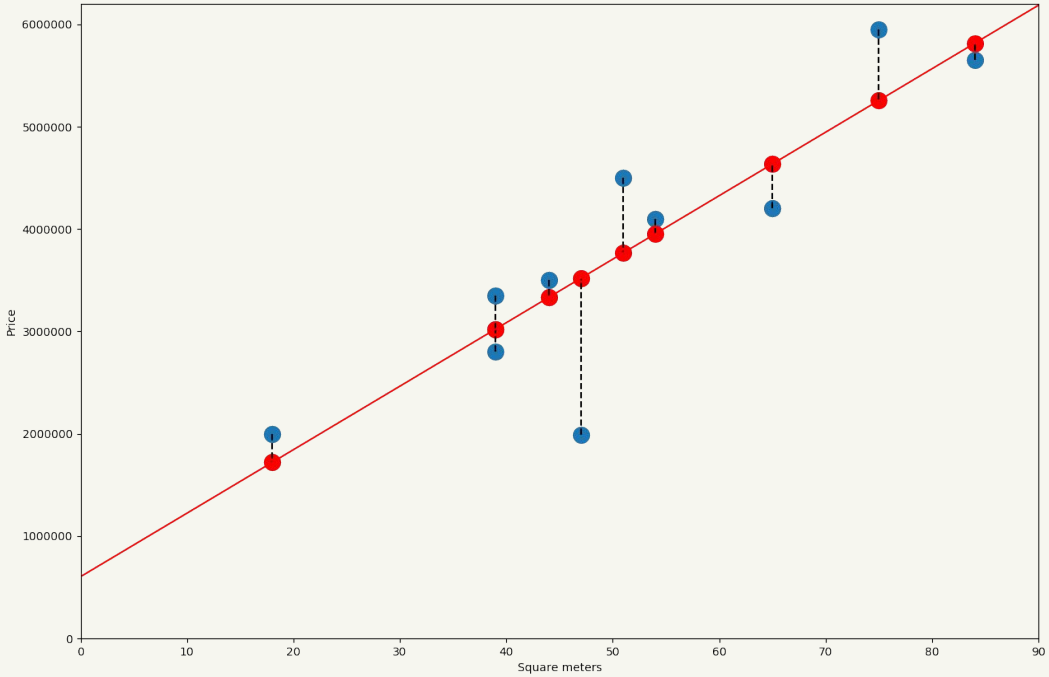
price = 62044(m^2)+602535



$$price = 62044(m^2)+602535$$

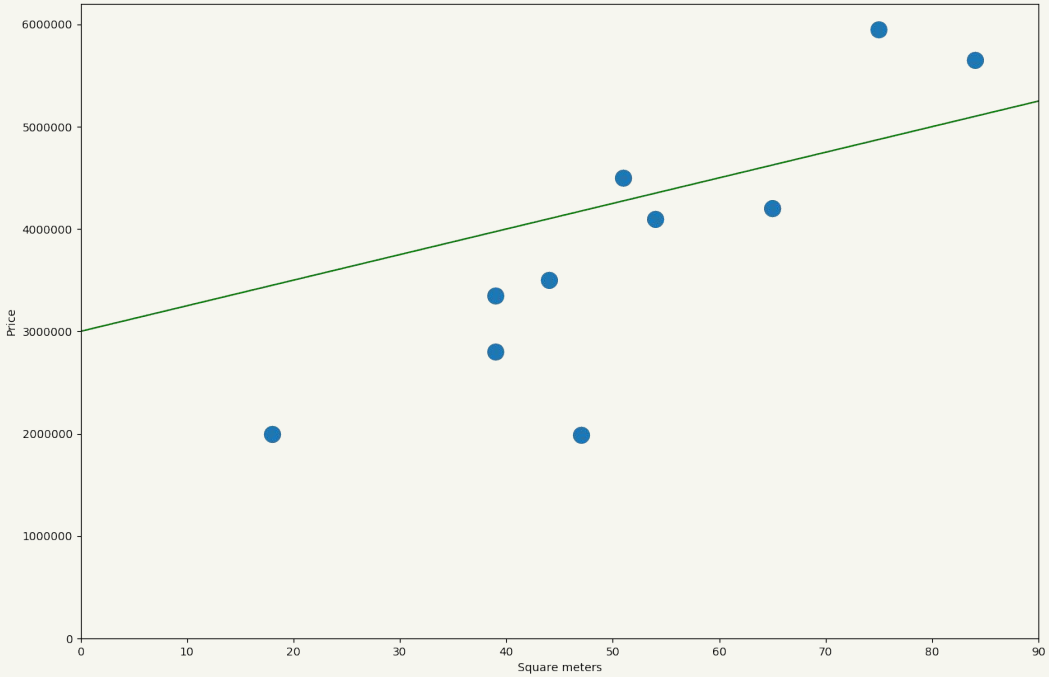


price = 62044(m^2)+602535

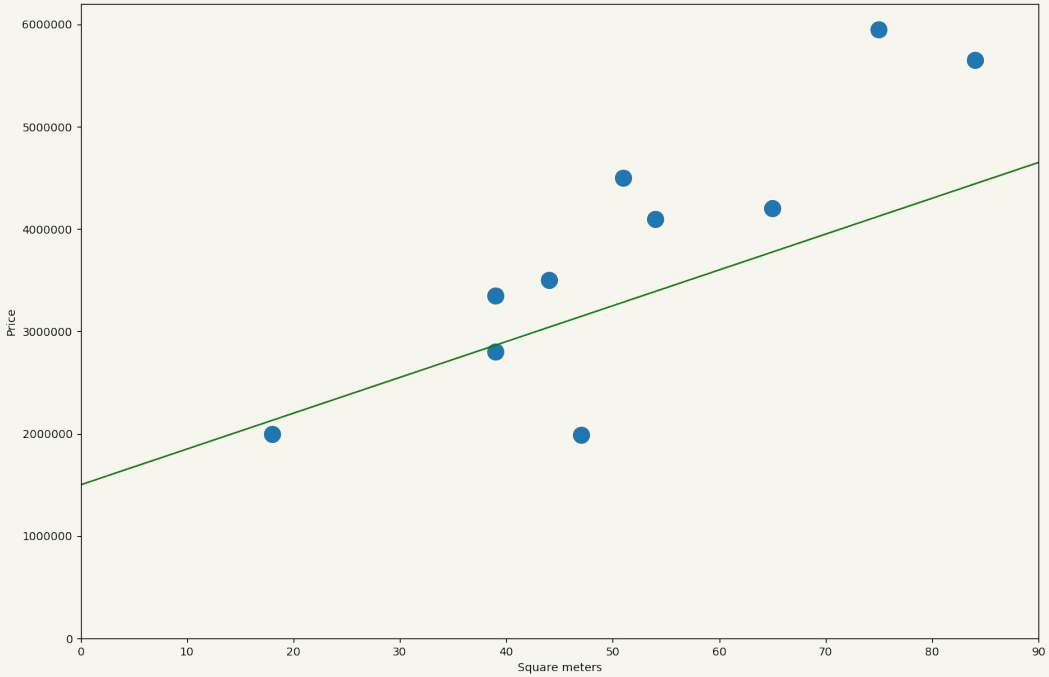


$$mse(f) = \frac{1}{N}\sum_{i=1}^{N}(y_i - f(x_i))^2$$

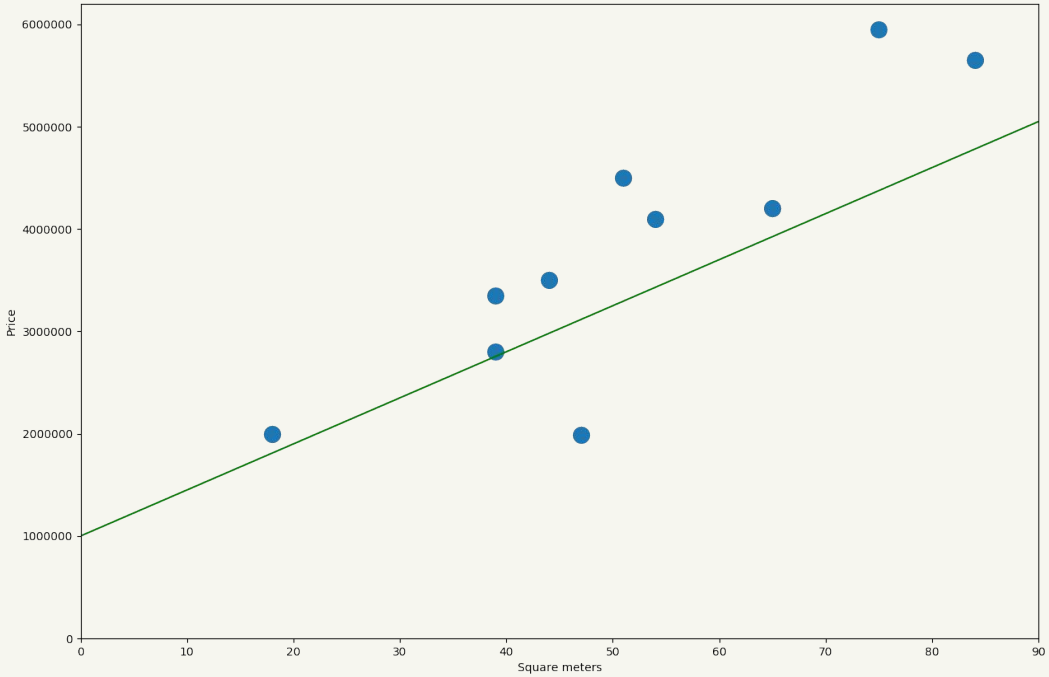
$$price = 25000(m^2)+3000000$$



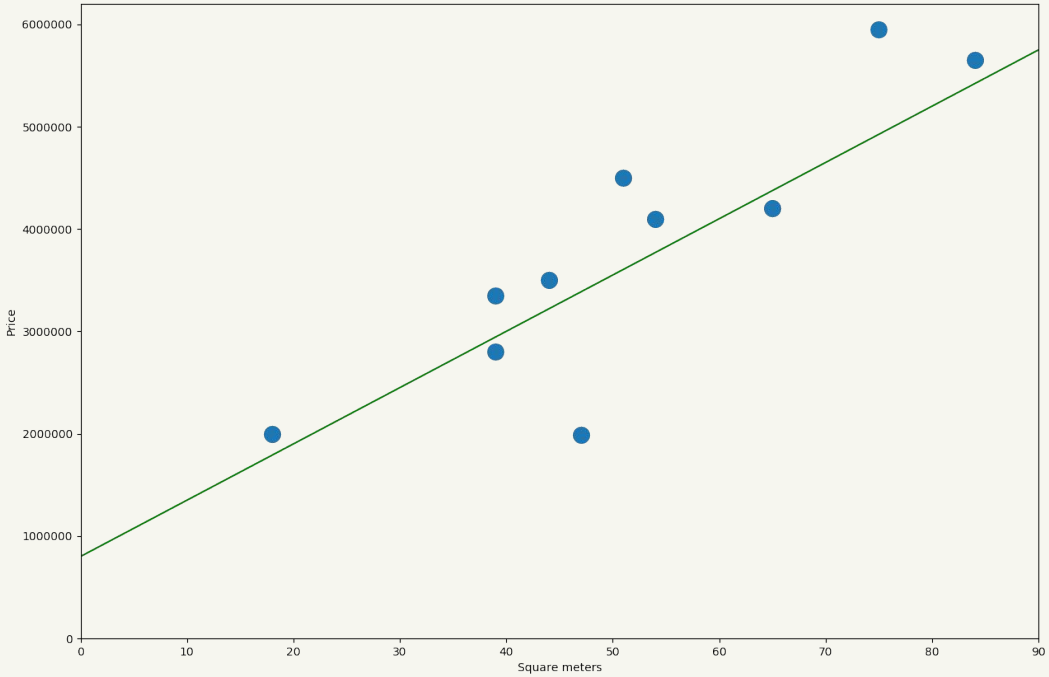
$$price = 35000(m^2)+1500000$$



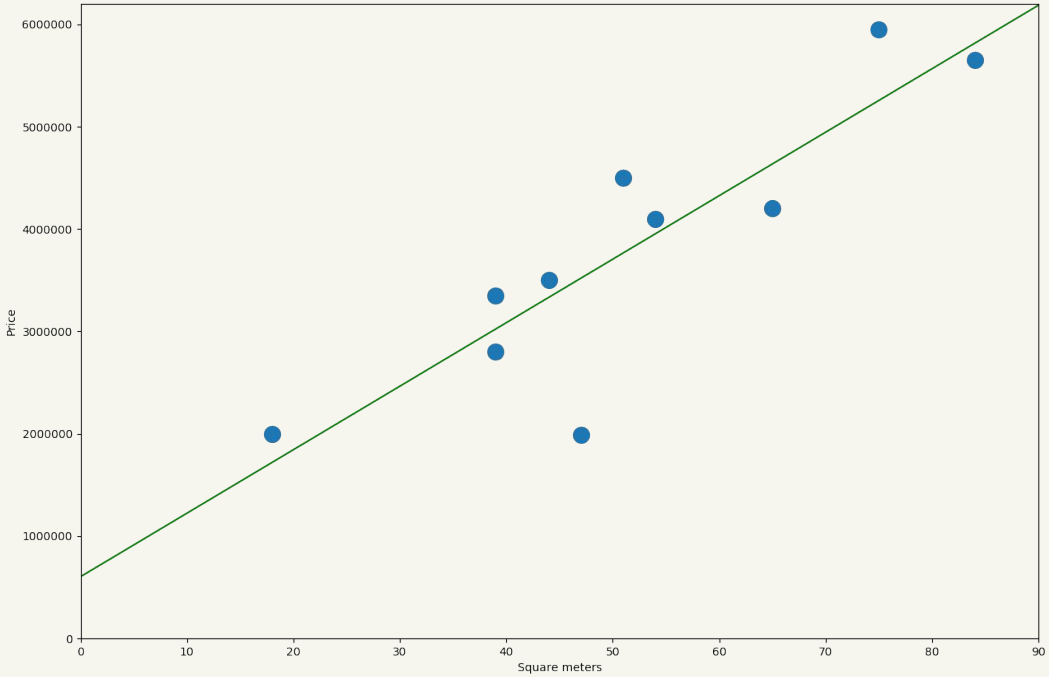
$$price = 45000(m^2)+1000000$$



$$price = 55000(m^2)+800000$$



$$price = 62044(m^2)+602535$$



price = 62044(m^2)+602535

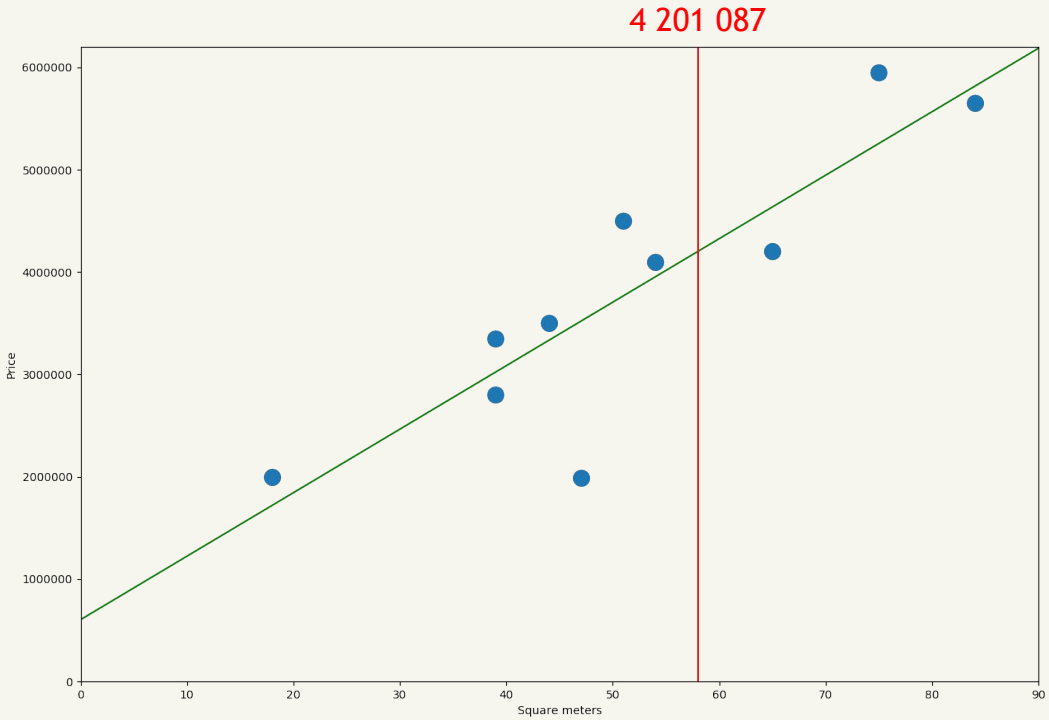
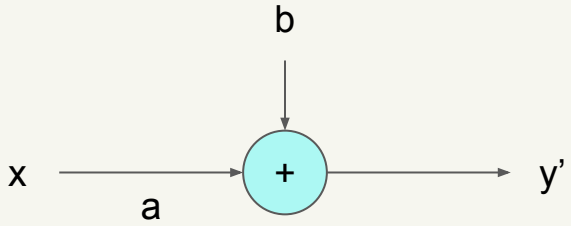
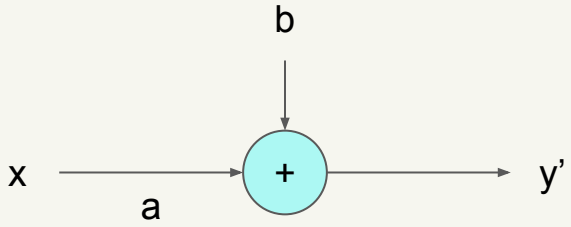


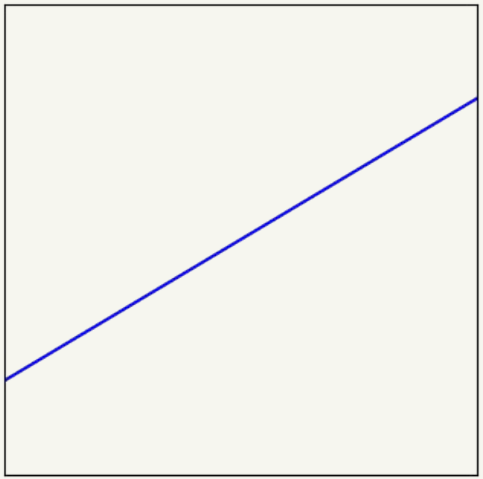
Image recognition

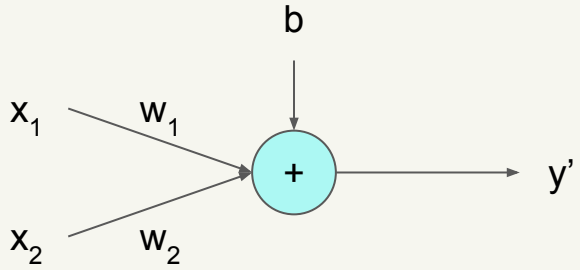
EPIGRAM MEDTECH

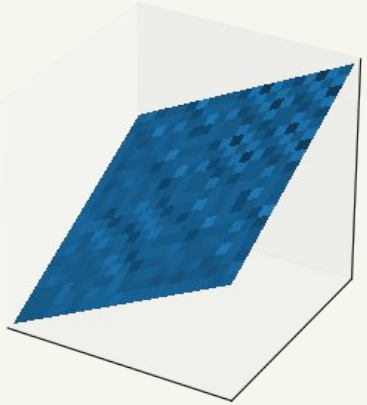


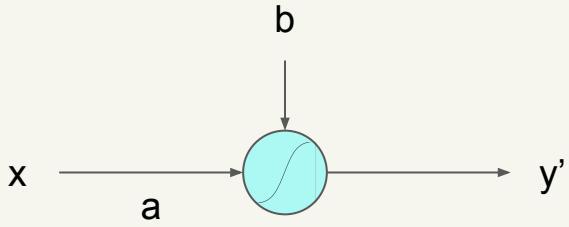
$$y' = ax + b$$

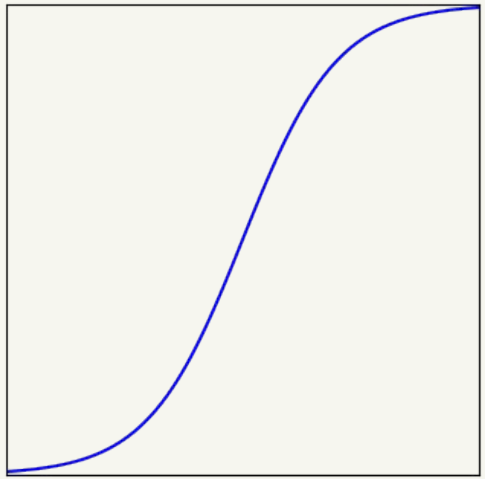

b
x
+
y’
a

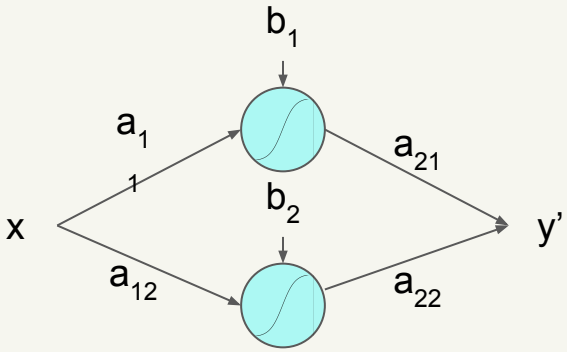


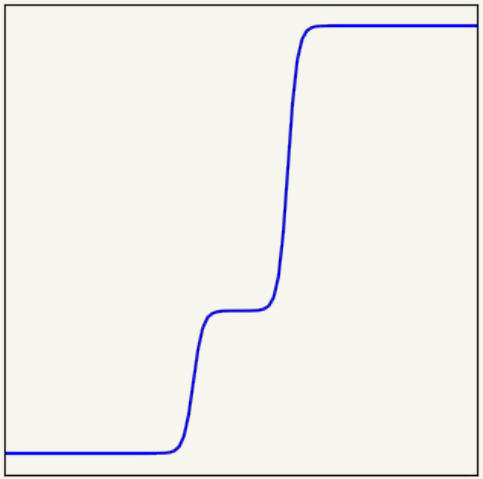

b
x1
w1
+
y'
x2
w2

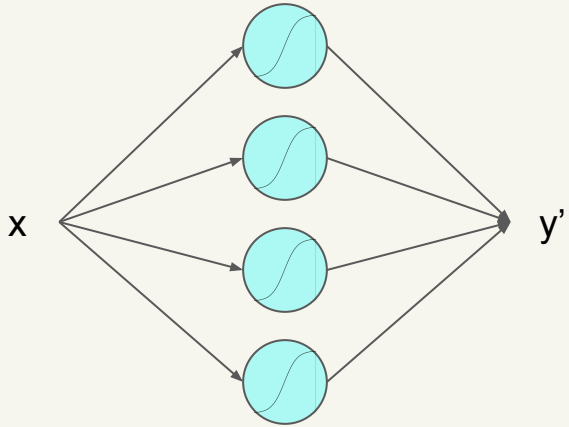


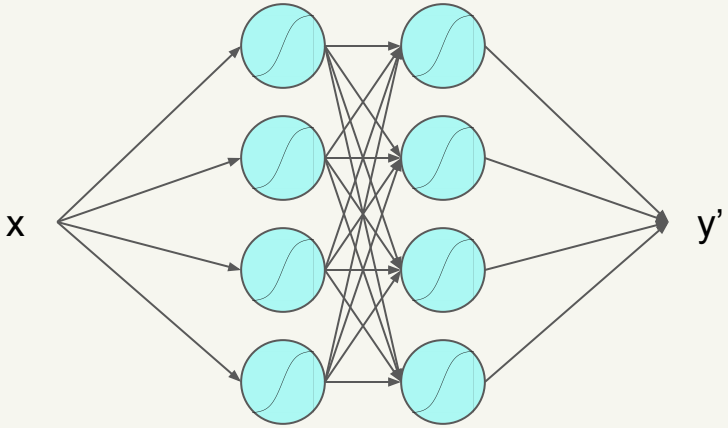

b
x
a
y’

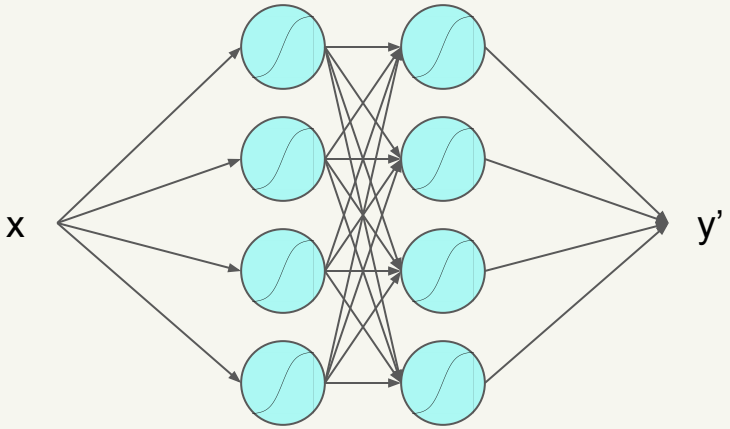



b1
a1
1
a21
x
b2
y'
a12
a22

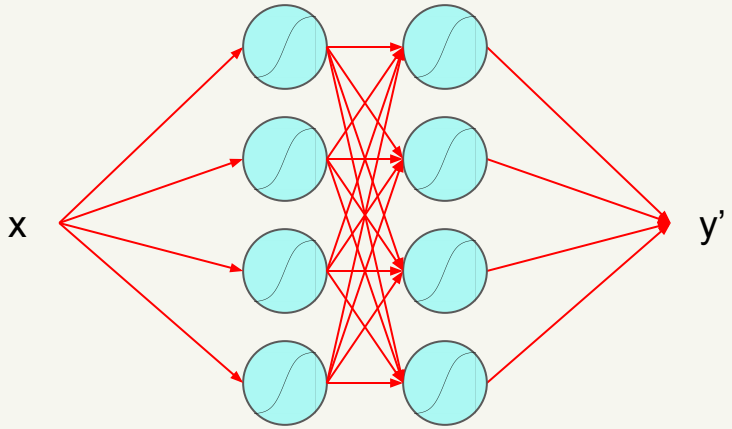



x
y’

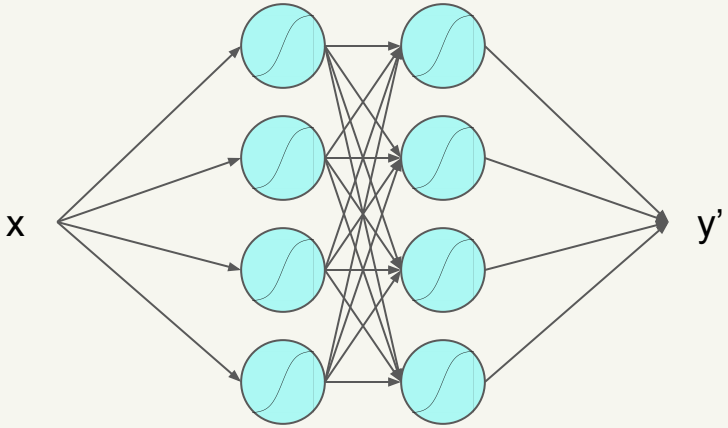

x
y’

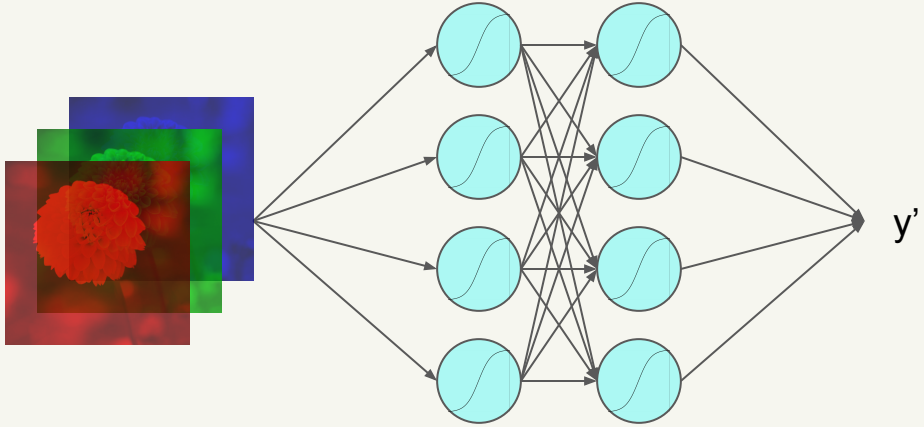


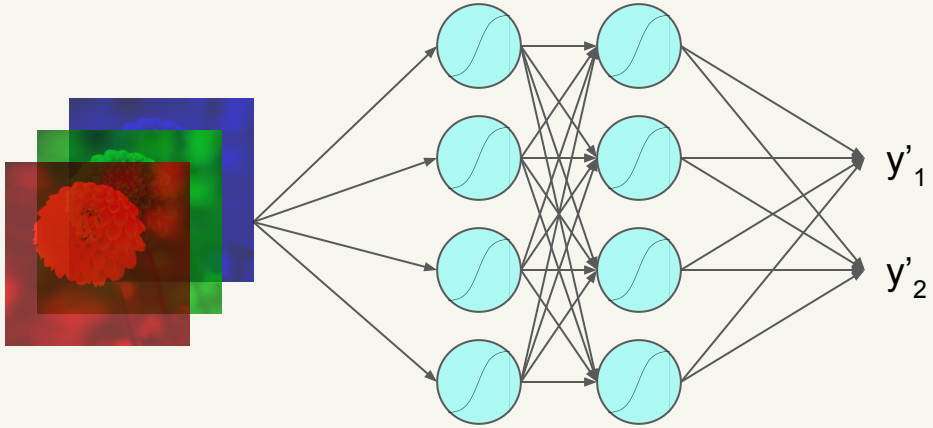
loss(y, y’)

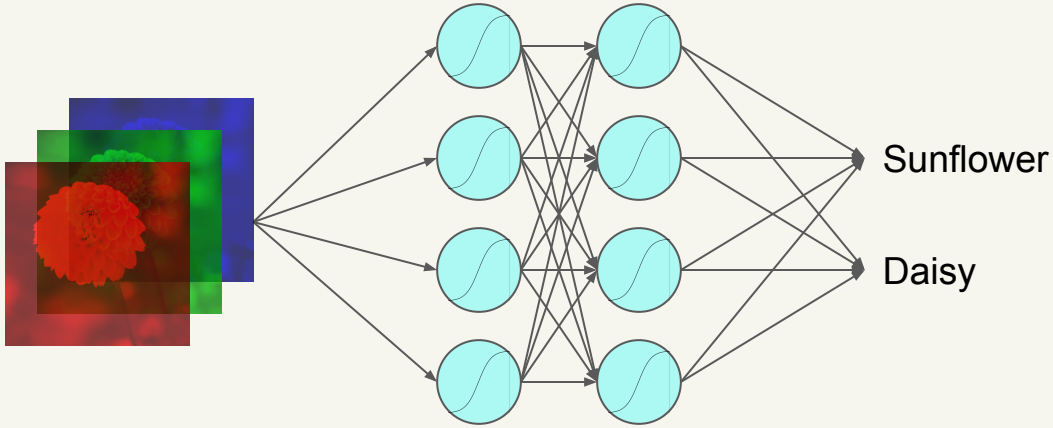


loss(y, y')

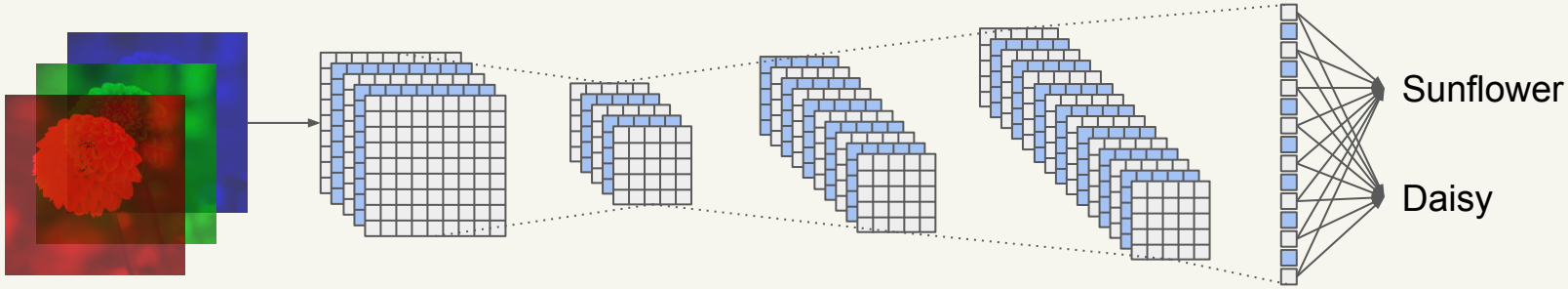

x
y’

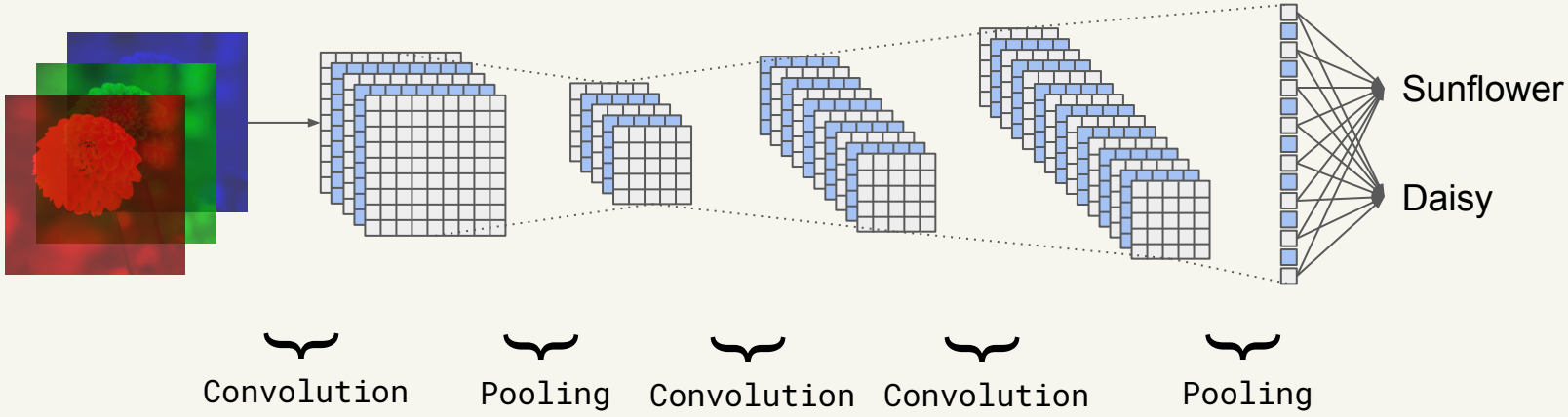

y’

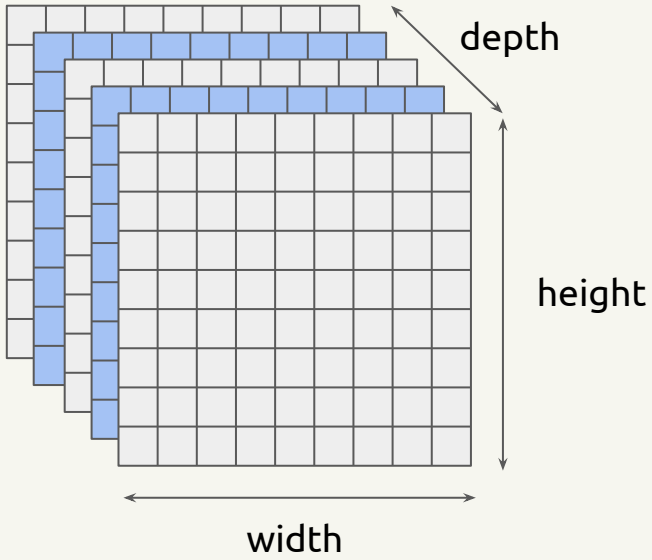

y'1
y'2

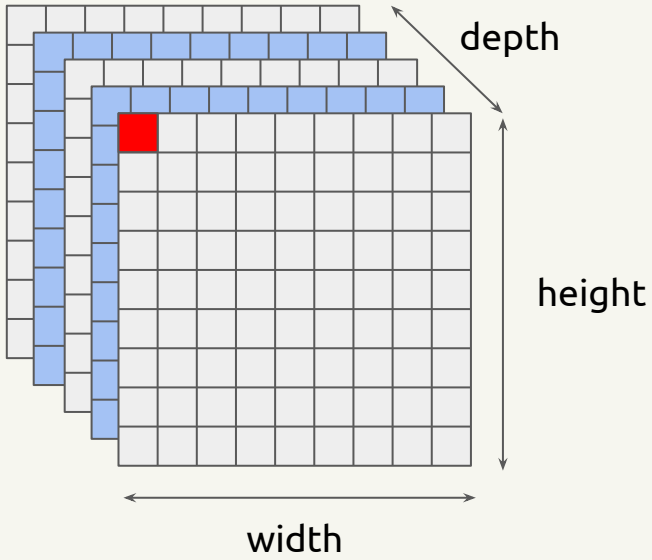

Sunflower
Daisy

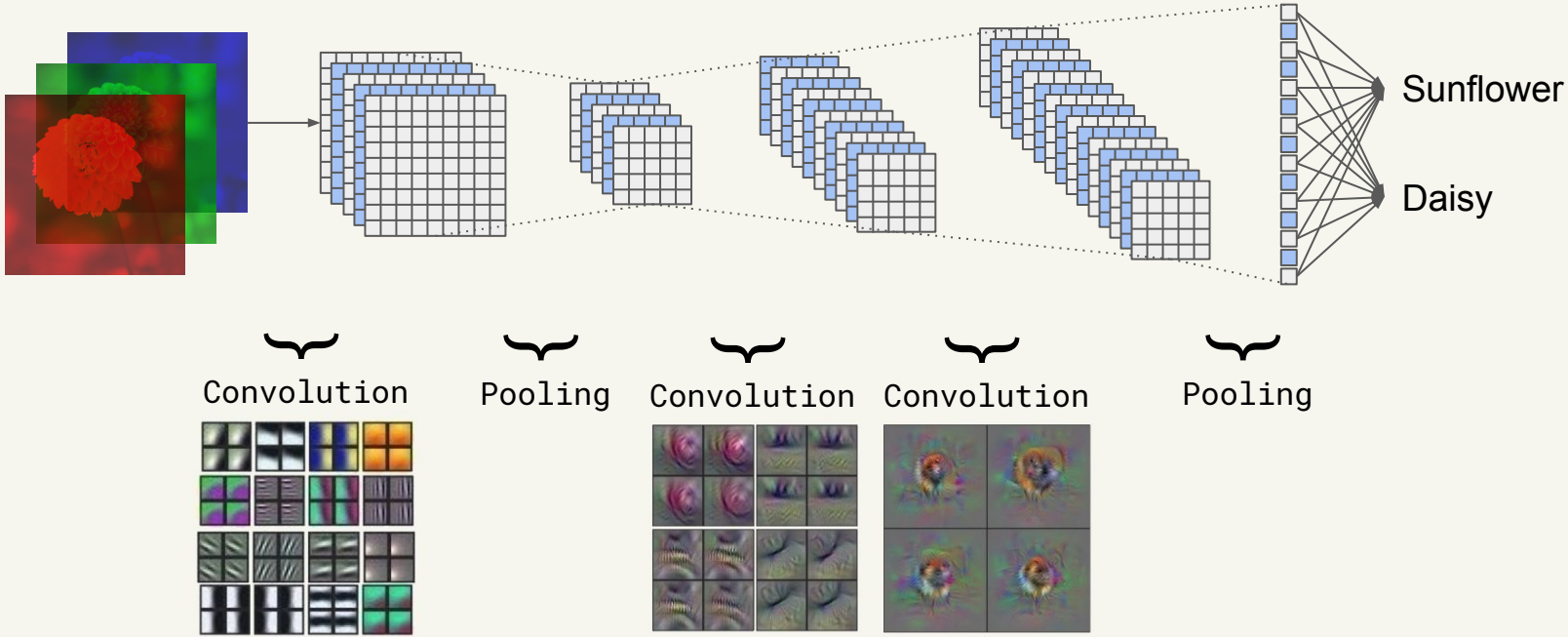
Convolutional Neural Nets

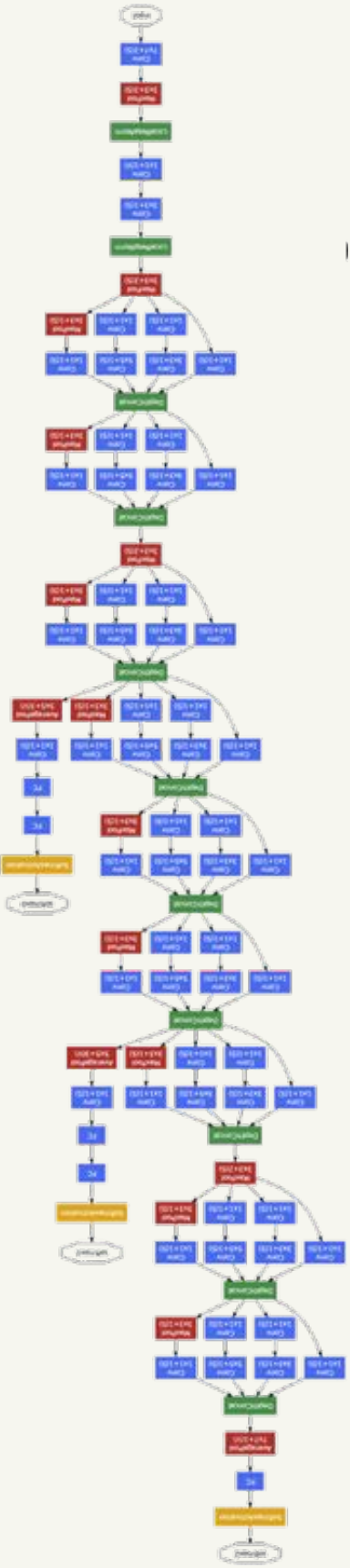

Sunflower
Daisy


Sunflower
Daisy
Convolution
Pooling
Convolution
Convolution
Pooling


depth
height
width

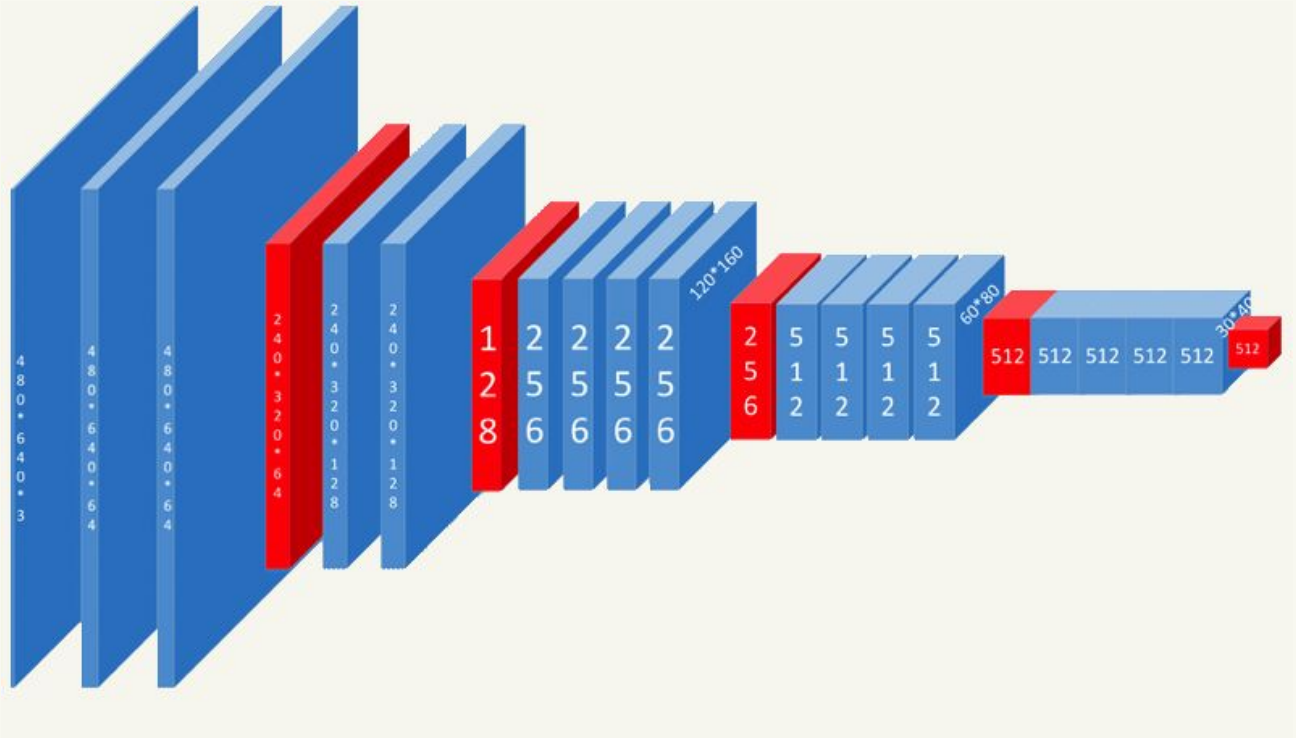

depth
height
width


Sunflower
Daisy
Convolution
Pooling
Convolution
Convolution
Pooling

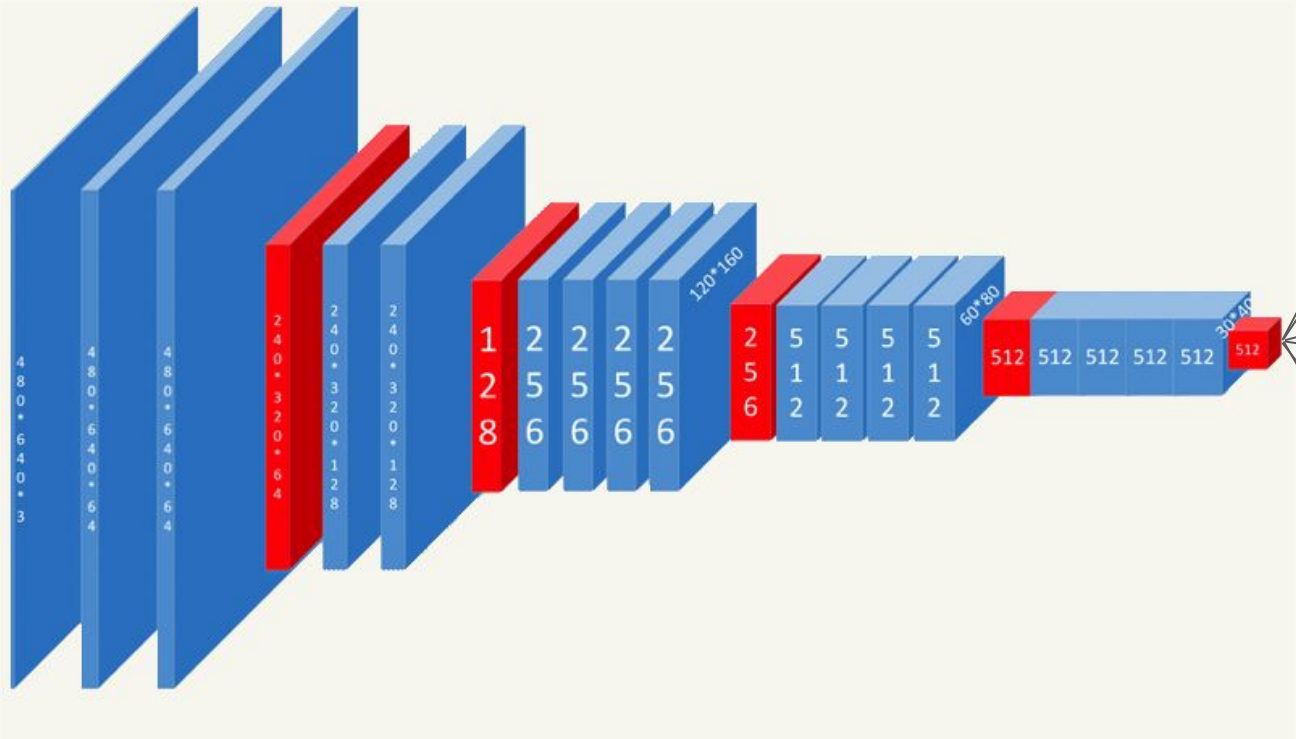


Transfer learning

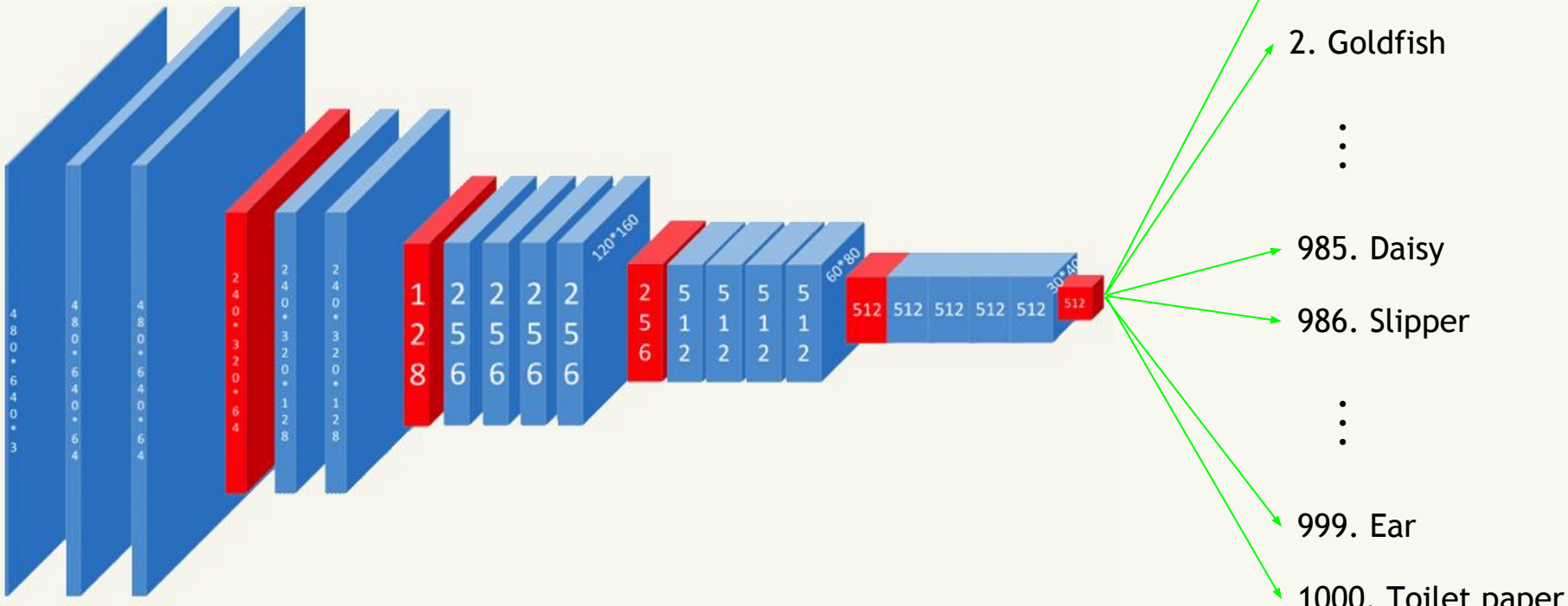
EPIGRAM MEDTECH

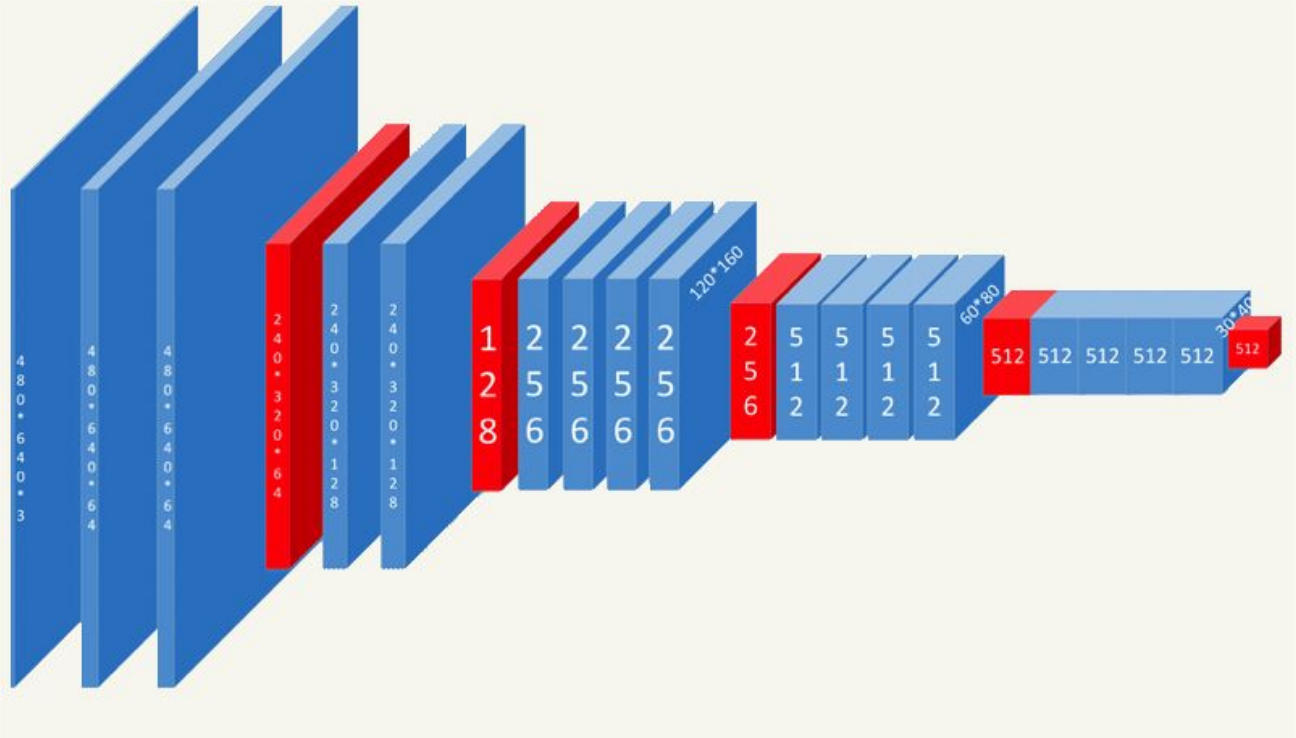

480*640*3
480*640*64
480*640*64
240*320*64
240*320*128
240*320*128
128
256
256
256
256
120*160
256
512
512
512
512
512
60*80
512
512
512
512
512
30*40
512

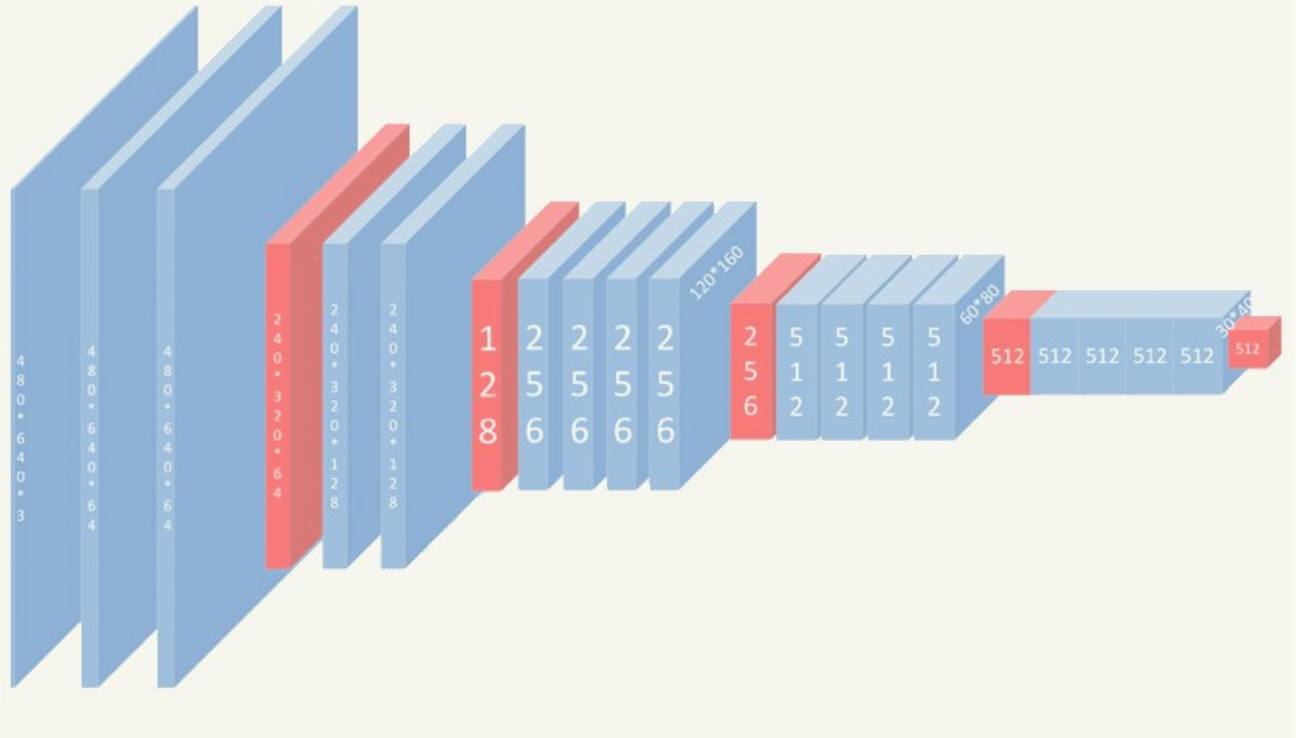

IMAGENET

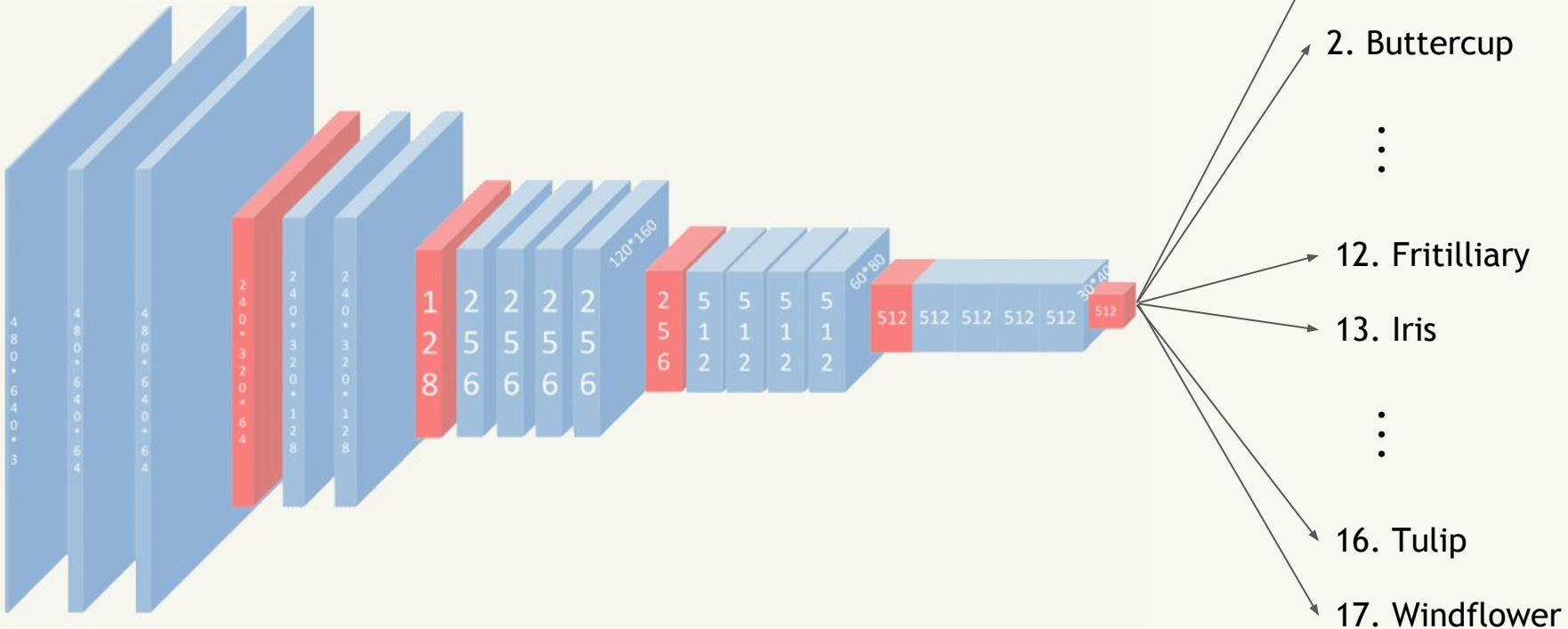


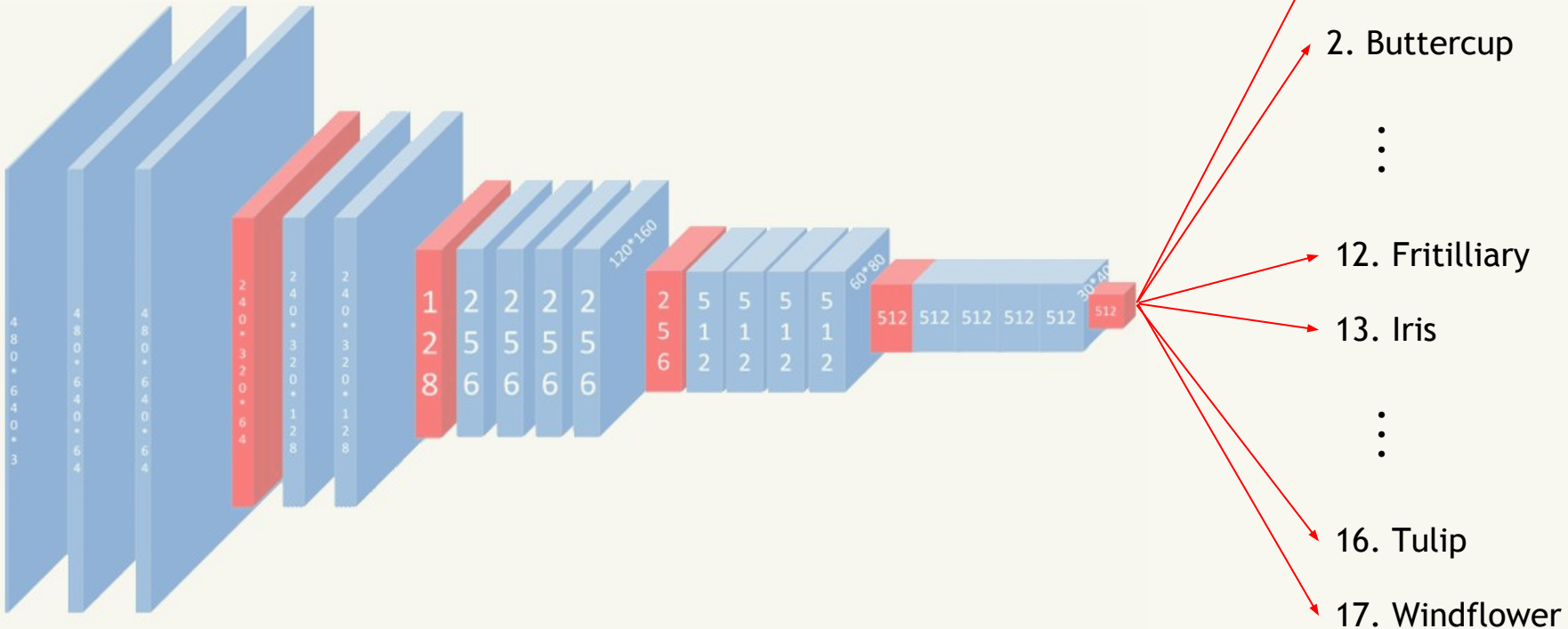
1. Tench
2. Goldfish
⋮
985. Daisy
986. Slipper
⋮
999. Ear
1000. Toilet paper


480*640*3
480*640*64
480*640*64
240*320*64
240*320*128
240*320*128
128
256
256
256
256
120*160
256
512
512
512
512
60*80
512
512
512
512
512
30*40
512
1. Tench
2. Goldfish
985. Daisy
986. Slipper
999. Ear
1000. Toilet paper


480*640*3
480*640*64
480*640*64
240*320*64
240*320*128
240*320*128
128
256
256
256
256
120*160
256
512
512
512
512
512
60*80
512
512
512
512
512
30*40
512

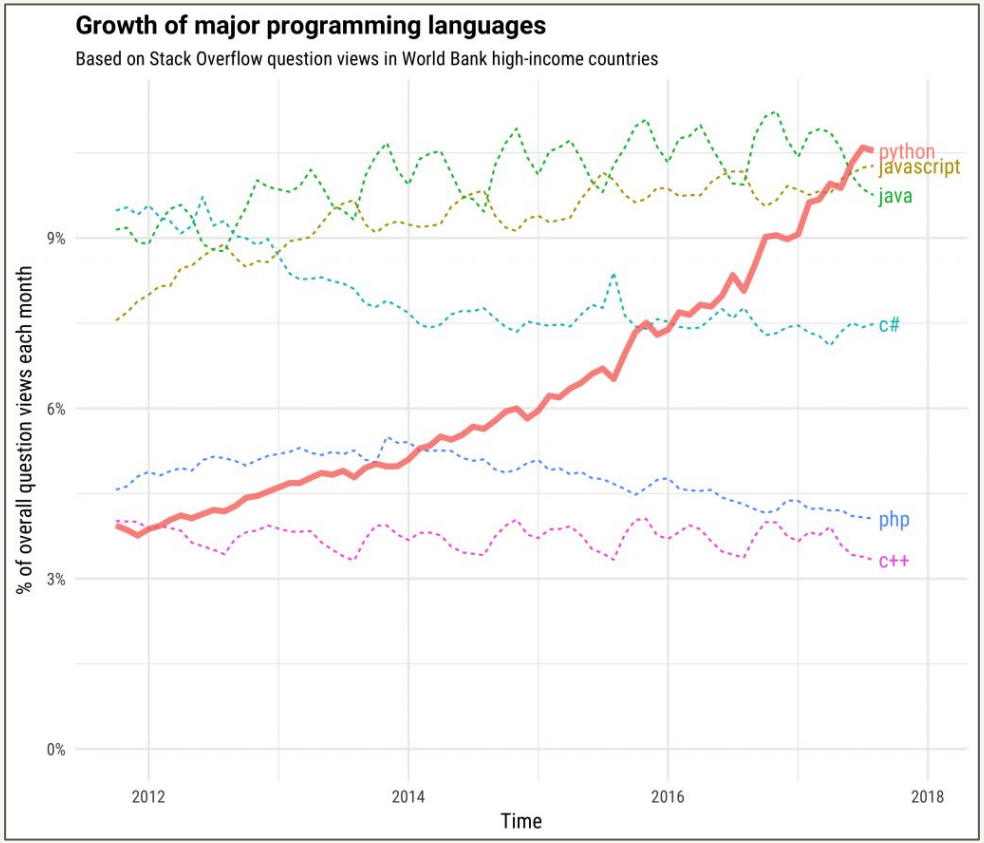

480*640*3
480*640*64
480*640*64
240*320*64
240*320*128
240*320*128
128
256
256
256
256
120*160
256
512
512
512
512
60*80
512
512
512
512
512
30*40
512

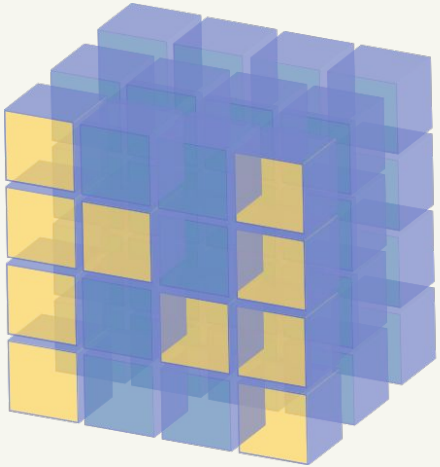

480*640*3
480*640*64
480*640*64
240*320*64
240*320*128
240*320*128
128
256
256
256
256
120*160
256
512
512
512
512
512
512
60*80
512
512
512
512
512
30*40
512
1. Bluebell
2. Buttercup
12. Fritilliary
13. Iris
16. Tulip
17. Windflower


480*640*3
480*640*64
480*640*64
240*320*64
240*320*128
240*320*128
128
256
256
256
256
120*160
256
512
512
512
512
512
60*80
512
512
512
512
512
30*40
512
1. Bluebell
2. Buttercup
12. Fritilliary
13. Iris
16. Tulip
17. Windflower

Flower classification

EPIGRAM MEDTECH


Growth of major programming languages
Based on Stack Overflow question views in World Bank high-income countries
% of overall question views each month
9%
6%
3%
0%
2012
2014
2016
2018
Time
python
javascript
java
c#
php
c++



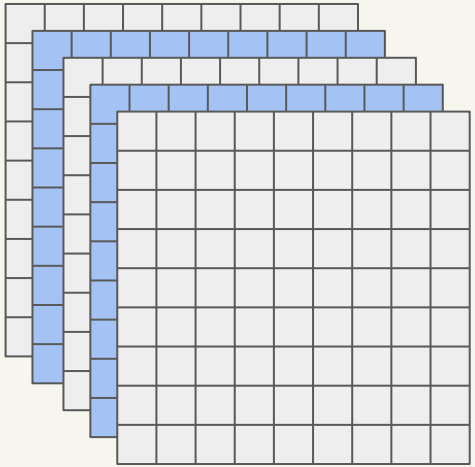
NumPy

TensorFlow

$$y = ax + b$$

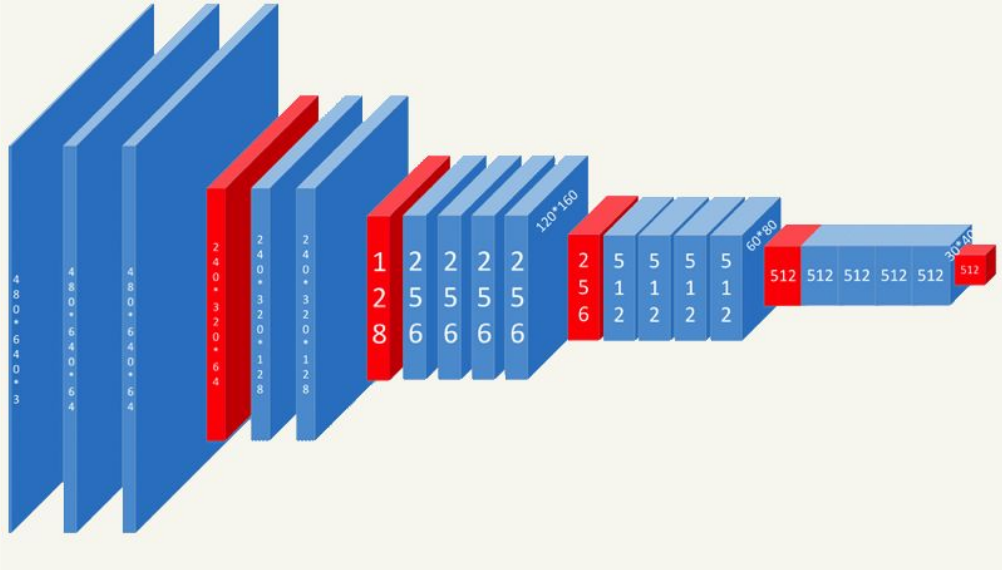
```
x = tf.placeholder(
    dtype=tf.float32,
    shape=(4, 4)
)
a = tf.Variable(
    initial_value=tf.zeros((4, 4)),
    dtype=tf.float32,
    trainable=True
)
b = tf.Variable(
    initial_value=tf.zeros((4, 4)),
    dtype=tf.float32,
    trainable=True
)

y = tf.add(tf.matmul(x, a), b)
```


K
Keras



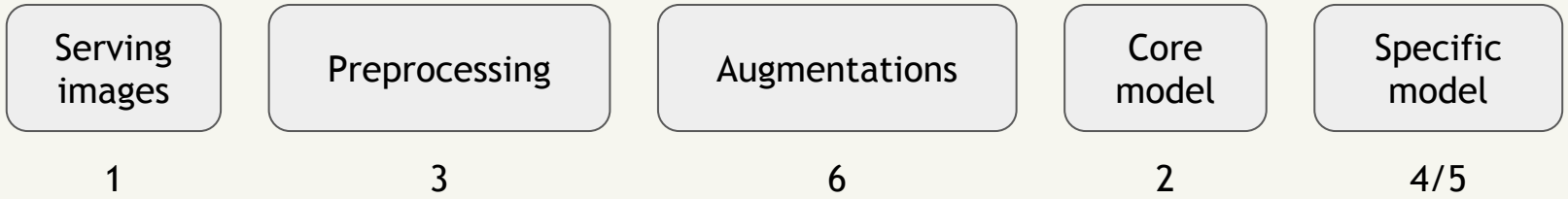
```
BATCH_SIZE = 4
IMAGE_SIZE = (256, 256, 3)

inputs = tf.placeholder(
    shape=(BATCH_SIZE,) + IMAGE_SIZE,
    dtype=tf.float32
)

weight_shape = tf.stack([5, 5, 3, 16])
weight_initializer = tf.random_normal(
    shape=weight_shape,
    stddev=.03
)
weights = tf.Variable(
    weight_initializer,
    trainable=True,
)

bias_initializer = tf.zeros(16)
bias = tf.Variable(
    bias_initializer,
    trainable=True
)

conv = tf.nn.conv2d(inputs, weights,
    strides=[1, 1, 1, 1],
    padding='SAME'
)
conv = tf.nn.bias_add(conv, bias)
```

```
IMAGE_SIZE = (256, 256, 3)

inputs = Inputs(shape=IMAGE_SIZE)

conv = Conv2D(kernels=16, filters=(3, 3))(inputs)
```

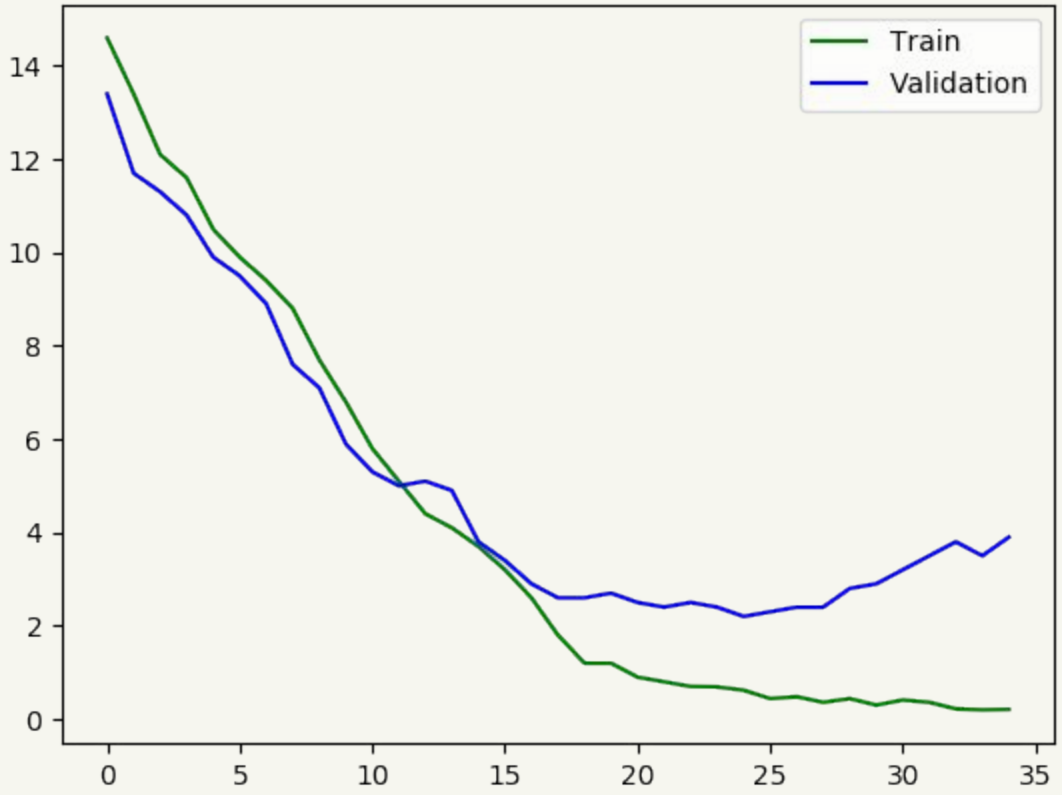



```
from keras.applications.vgg19 import VGG19

model = VGG19()
```

```
from keras.applications.vgg19 import VGG19

model = ... # Create a model

model.summary() # Print the structure of the model

model.fit() # Train the model

model.evaluate() # Evaluate performance on a validation set

model.predict() # Run predictions on new data

model.save() # Save the model to file

model.load() # Load the model into memory
```


Serving images
Preprocessing
Augmentations
Core model
Specific model
1
3
6
2
4/5

Labels:	Indexes:	Onehot encoded:
‘bluebell’	0	[1, 0, 0, 0, 0, 0]
‘buttercup’	1	[0, 1, 0, 0, 0, 0]
‘colts_foot’	2	[0, 0, 1, 0, 0, 0]
‘cowslip’	3	[0, 0, 0, 1, 0, 0]
‘crocus’	4	[0, 0, 0, 0, 1, 0]
‘daffodil’	5	[0, 0, 0, 0, 0, 1]


Train
Validation
14
12
10
8
6
4
2
0
0
5
10
15
20
25
30
35

esten@epimed.ai

EPIGRAM
MEDTECH