Weekly Report

Wangwon Lee, 2019/05/25

This week

- **Training 2D CNN**
  - Our imagenet data
  - Fine tuning (simply)
- **Visualization**
  - Deconvolution Network (our data)

Next week

- **Visualization**
  - Apply another visualization method
  - Apply to 1D (our data)
- **Training 2D CNN**
  - Retrain cleaned data

Interesting and new finding

- Visualization
- Data Cleaning

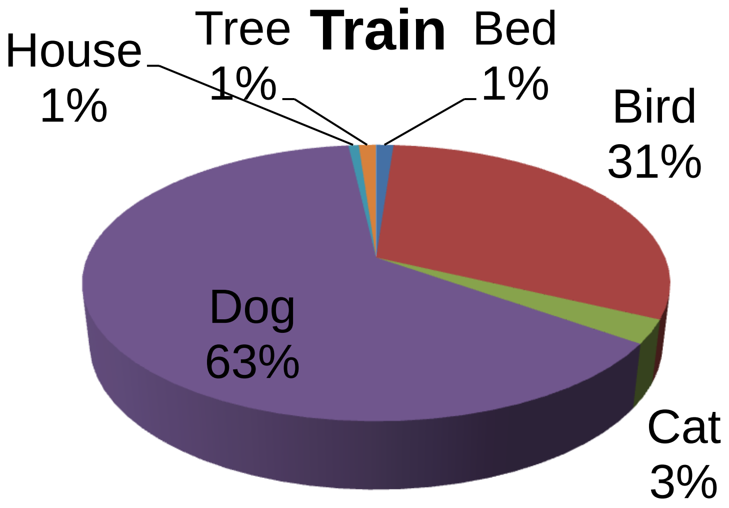
The aim of this month / Discussion

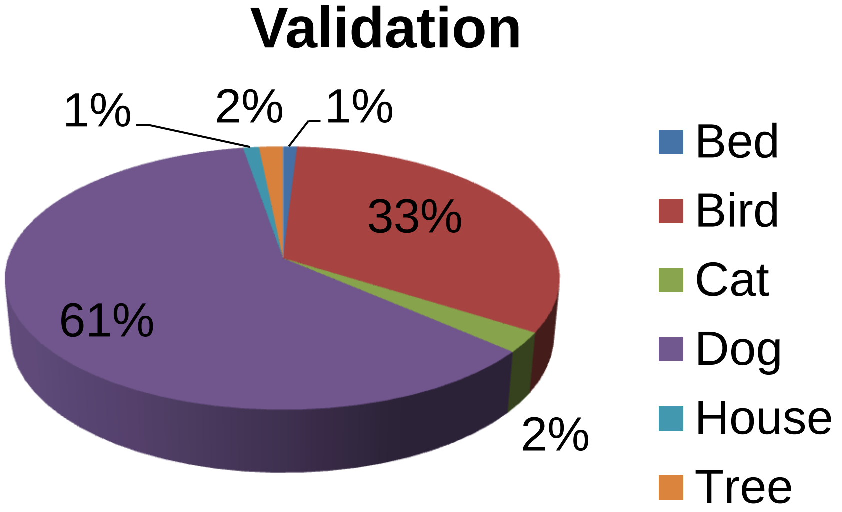
- **The aim of this month**: To study brain data.

Training 2D CNN - Data

- Data is very imbalanced
- Train: 235111, Val: 3183
- Training is based on undersampling

Label	Train	Val
Bed	2690	30
Bird	72641	1050
Cat	6500	78
Dog	149006	1940
House	1611	34
Tree	2663	51

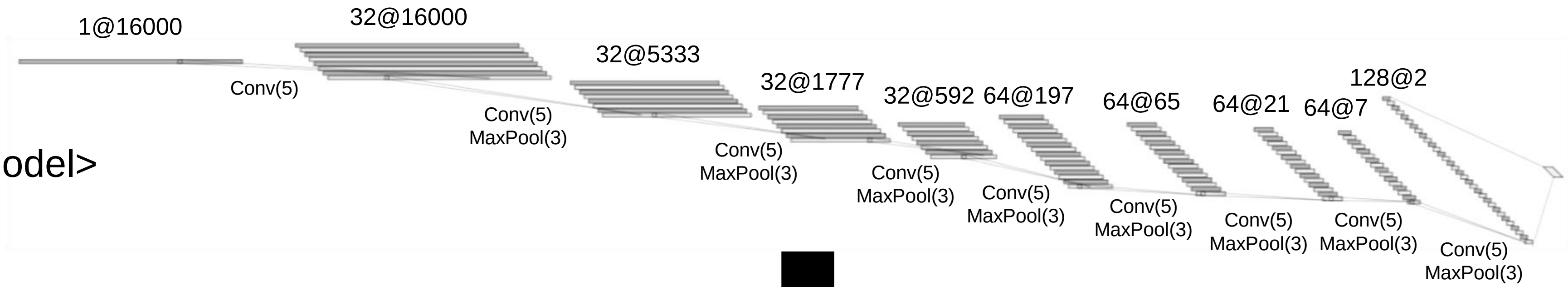




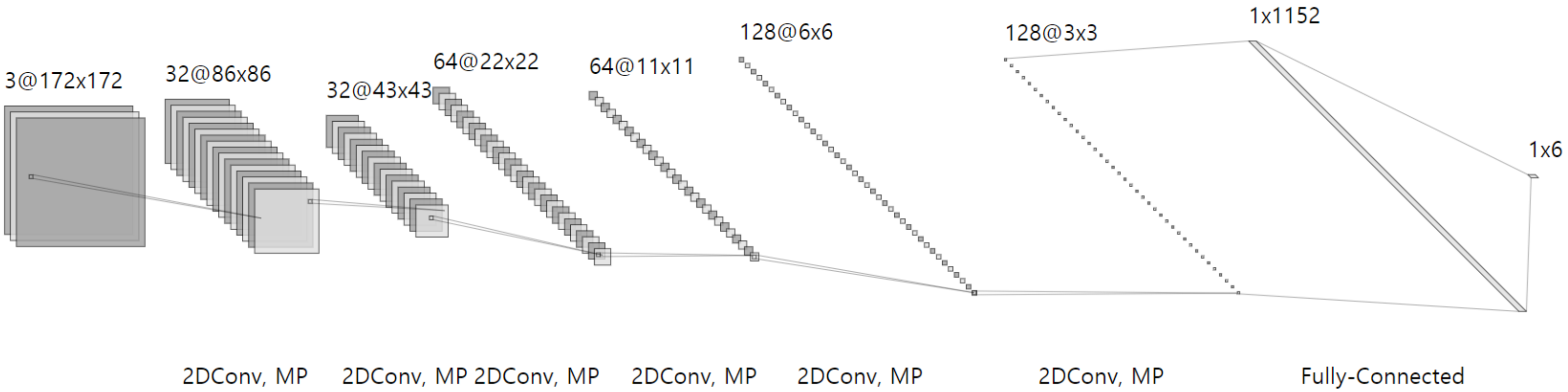
Training 2D CNN - Change

- Change (Because of input data’s shape)
  Input: 16000X1 → 172X172X3, Output: 16 → 6 ('bed', 'bird', 'cat', 'dog', 'house', 'tree')
  filter size: 1X5 → 5X5, pool size: 1X3 → 2X2, number of conv: 8 → 7(Best Fit)

<1D Model>



<2D Model>
(Ex. 6Conv)



Training 2D CNN - Result

- To find best fit model, Tuning the depth and start channel(32, 64)
- Most of case, 0.75 dropout rate show better performance than 0.5. But, To train more faster, Setting the dropout rate as 0.5.
- I think ‘CH32, DO0.5, 6 Conv’ model is best, for this task.

Architecture	CH32+DO0.5	Params	CH64+DO0.5	Params
1 CONV(5X5)	X	1,422,470	X	2,844,934
2 CONV(5X5)	X	383,078	X	817,350
3 CONV(5X5)	0.7338	265,190	0.8325	683,974
4 CONV(5X5)	0.8130	228,262	0.8998	814,918
5 CONV(5X5)	0.8014	414,374	0.9258	1,596,742
6 CONV(5X5)	0.9044	803,366	0.9242	3,193,926
7 CONV(5X5)	0.8881	1,622,054	0.9359	6,469,702

Training 2D CNN - Result

- In the best architecture, It seems pretty good.
  except confusing cat as dog……
- Maybe It's because the model is not complicated enough.

Actual class

Predict Class

```
[[  28    0    1    1    0    0]
 [   5  932   12   85    2   14]
 [   0    3   62   13    0    0]
 [   4   29   75 1831    0    1]
 [   0    0    0    1   32    1]
 [   0    1    0    1    2   47]]
```

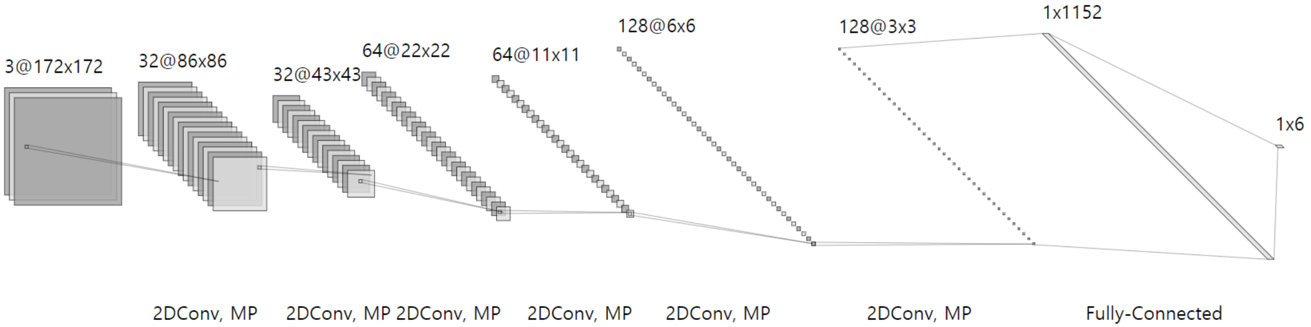
```
              precision    recall  f1-score   support

         bed       0.76      0.93      0.84        30
        bird       0.97      0.89      0.93      1050
         cat       0.41      0.79      0.54        78
         dog       0.95      0.94      0.95      1940
       house       0.89      0.94      0.91        34
        tree       0.75      0.92      0.82        51

weighted avg       0.93      0.92      0.93      3183
```

<Best Architecture>

(Acc: 0.9044)

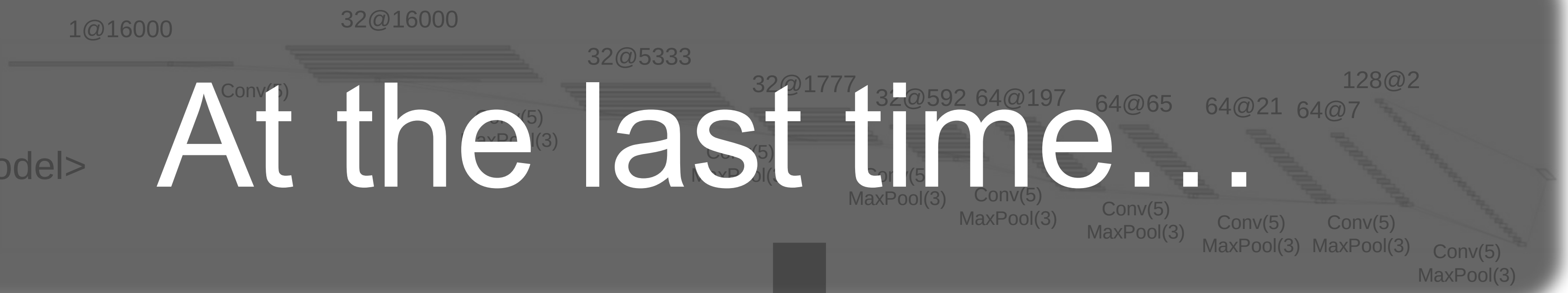


Training 2D-Model for visualization (MNIST)

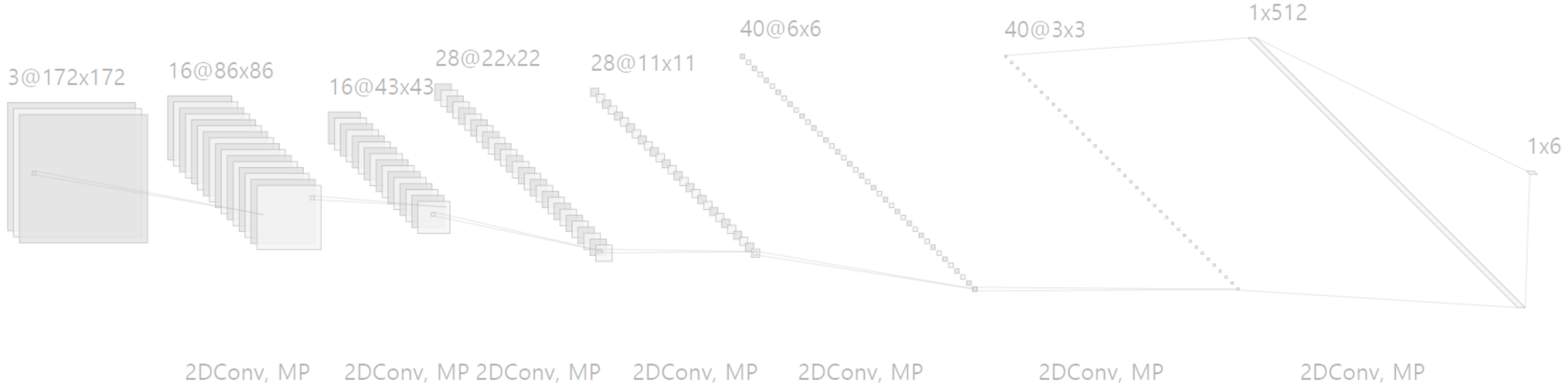
- Change (Because of input data’s shape)
  Input: 16000X1 → 28X28X1, Output: 16 → 10 (0 ~ 9)
  filter size: 1X5 → 3X3, pool size: 1X3 → 2X2, number of conv layer: 8 → 3
- Train: 40000, Val: 20000, Test:10000

At the last time…

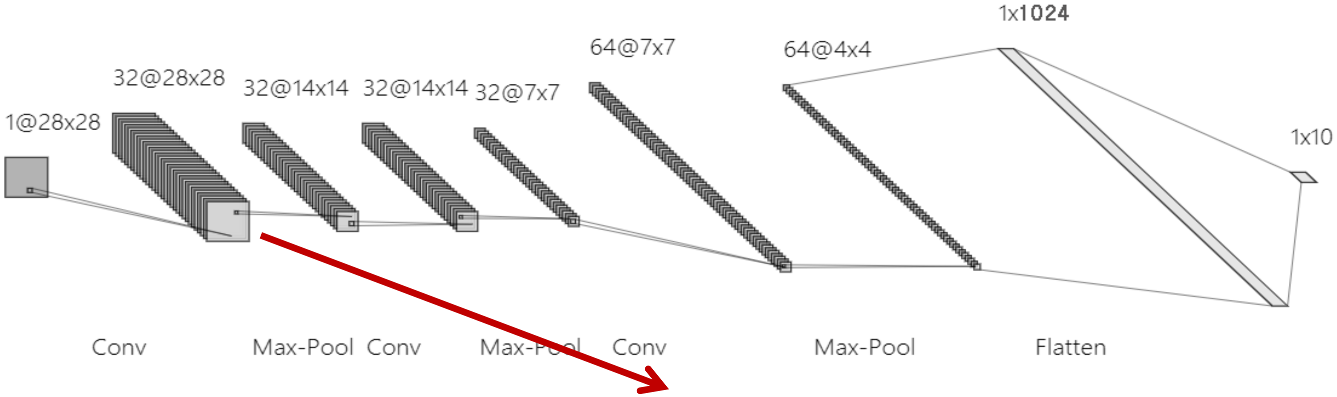
<1D Model>

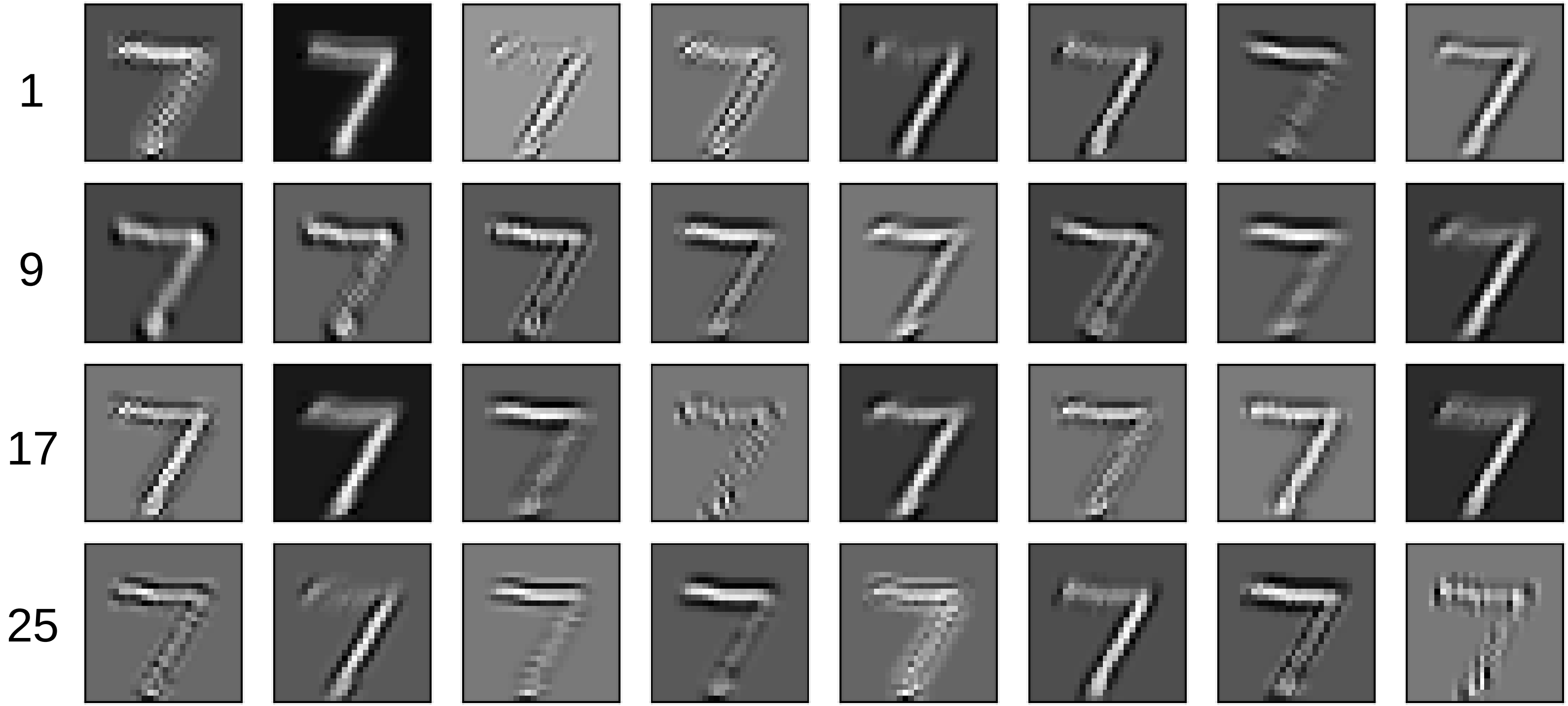


<2D Model>

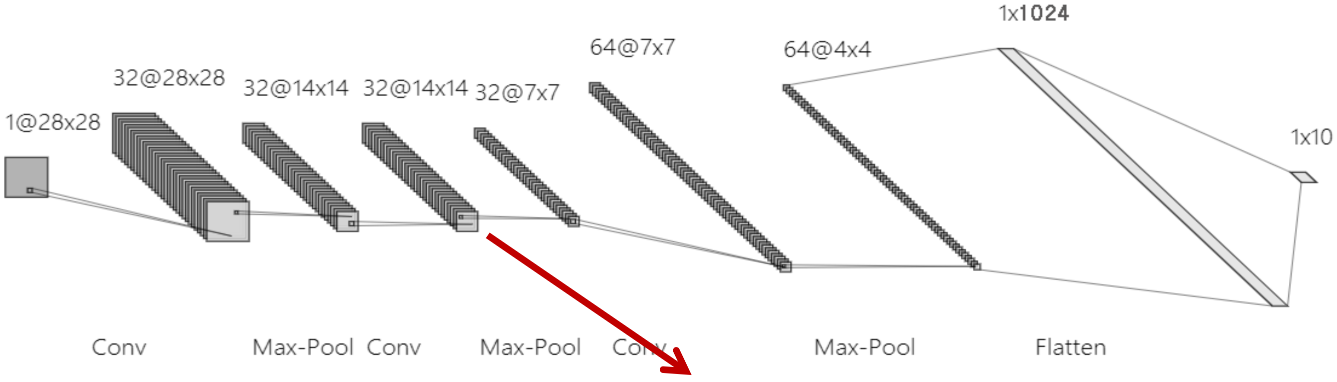


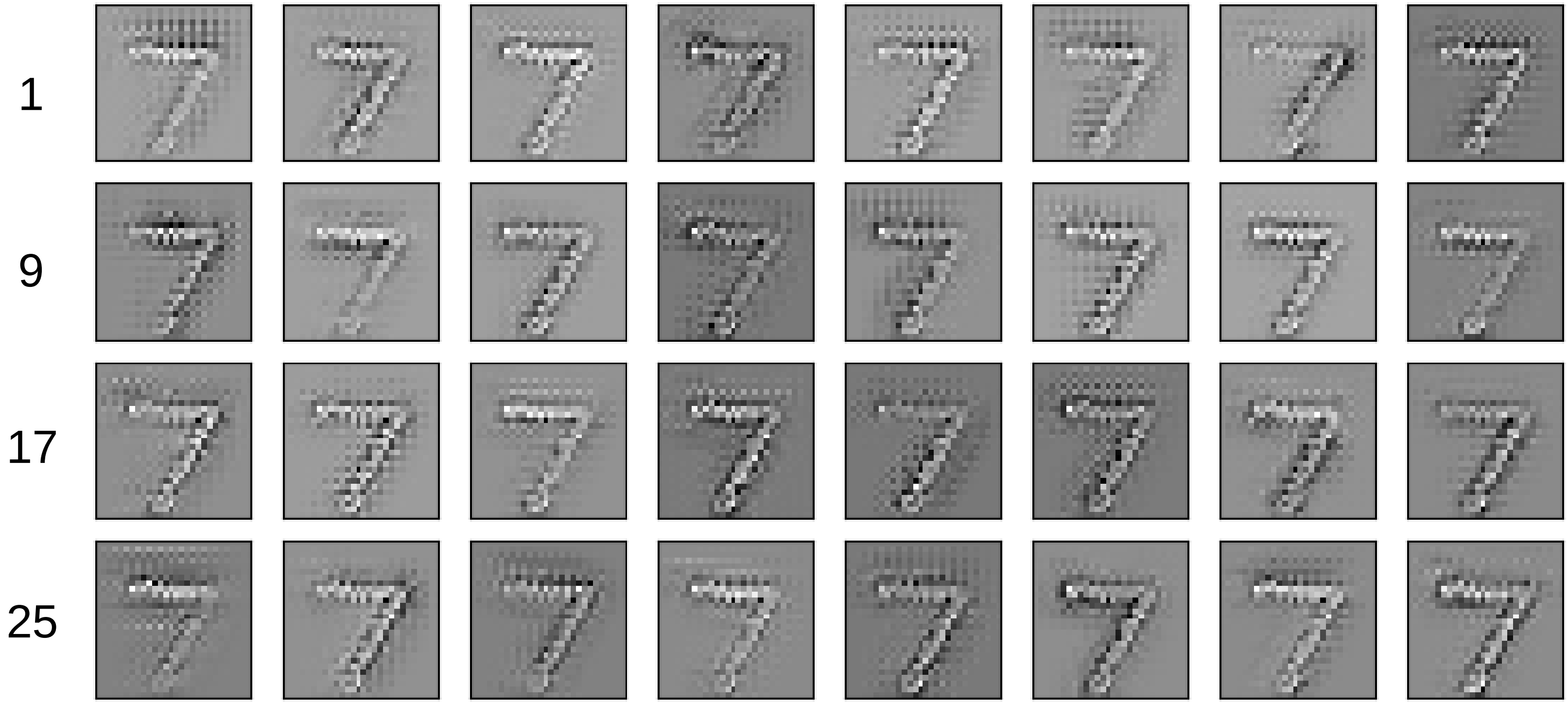
Visualization – mnist (1^{st} Conv)



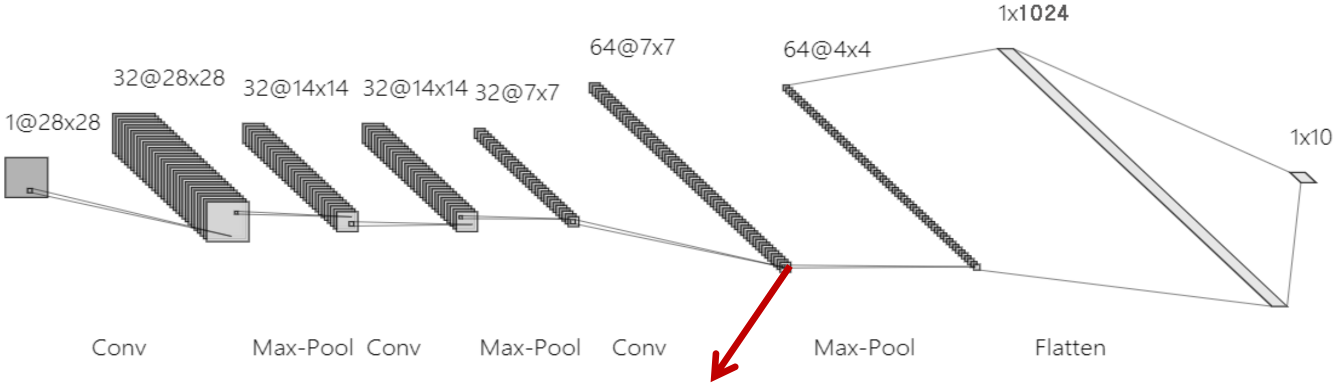


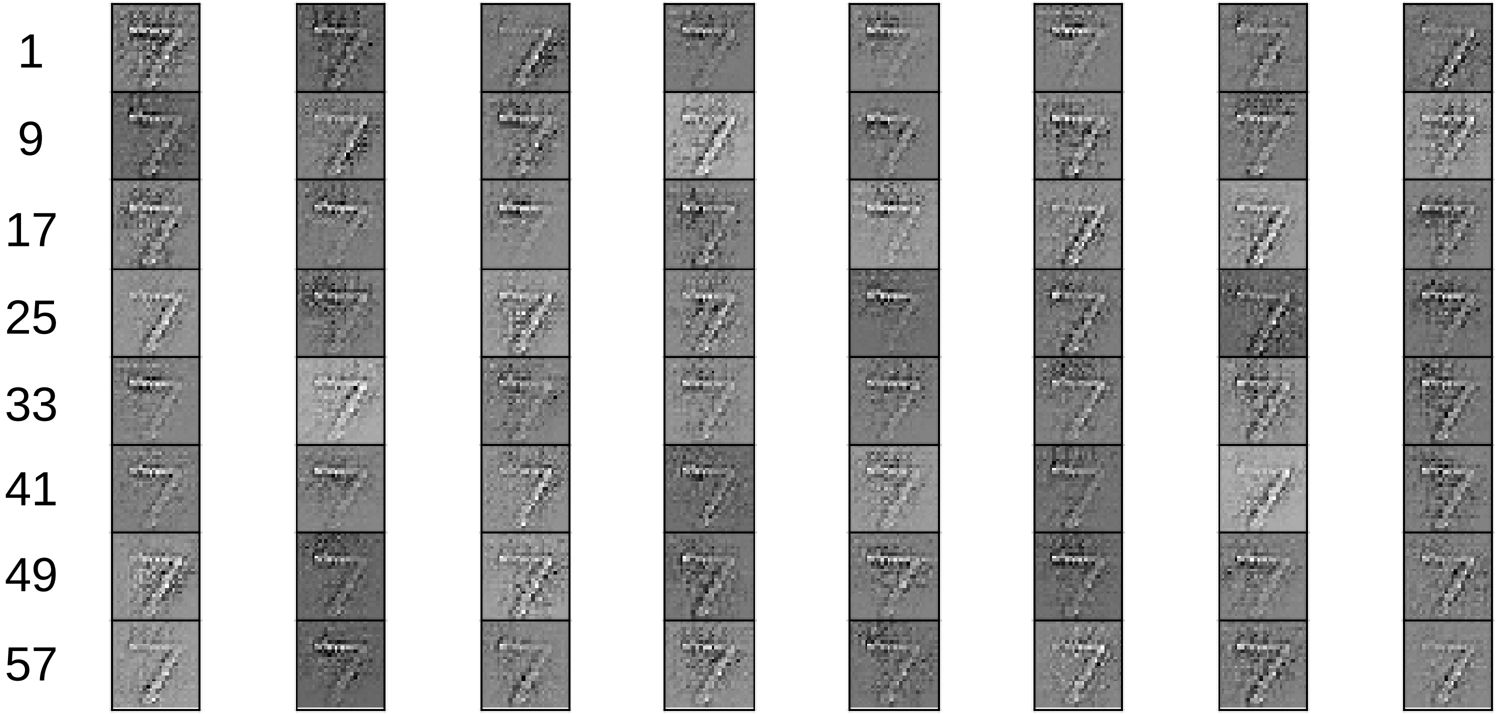
Visualization – mnist (2^{nd} Conv)





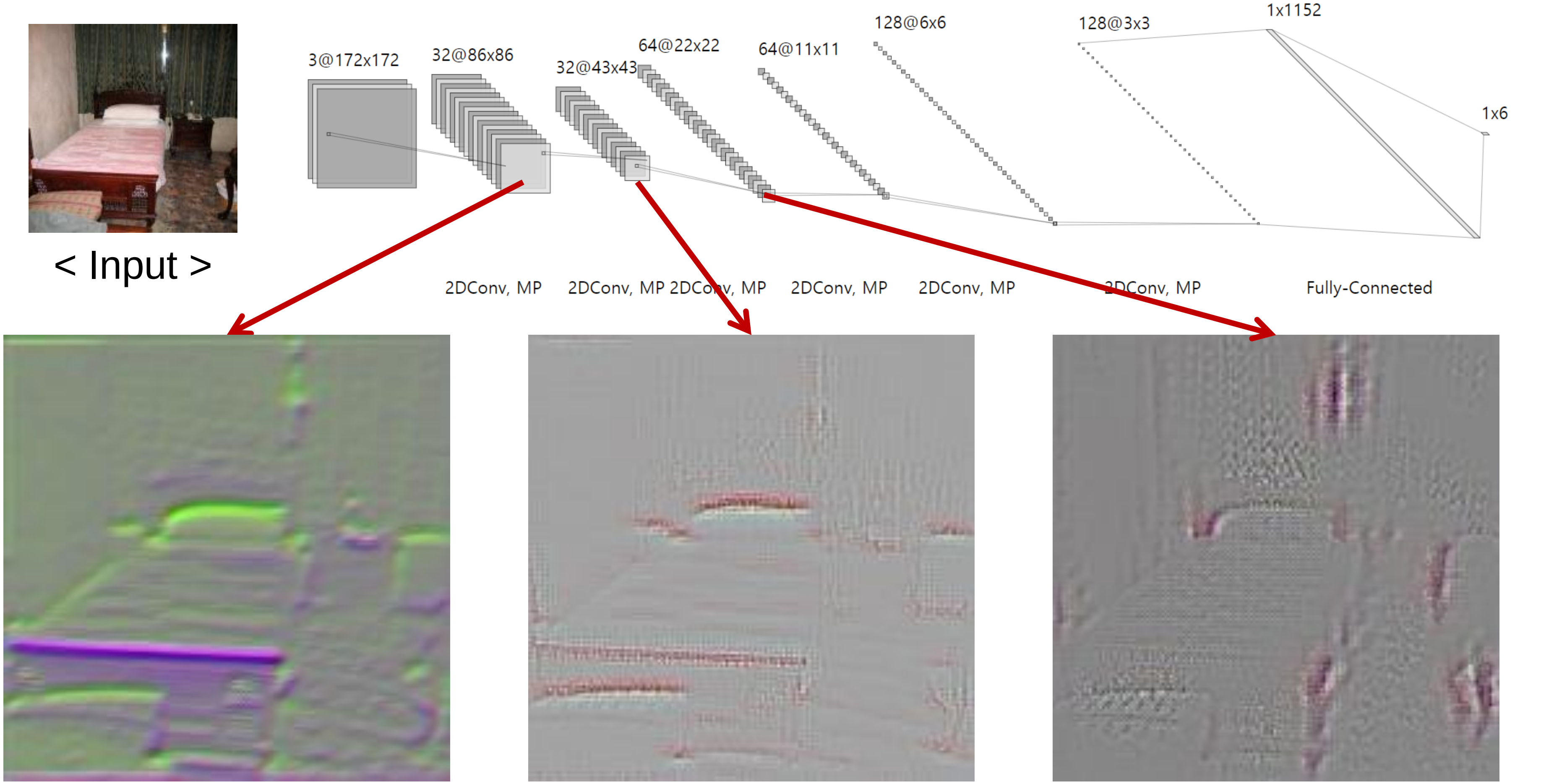
Visualization – mnist (3rd Conv)





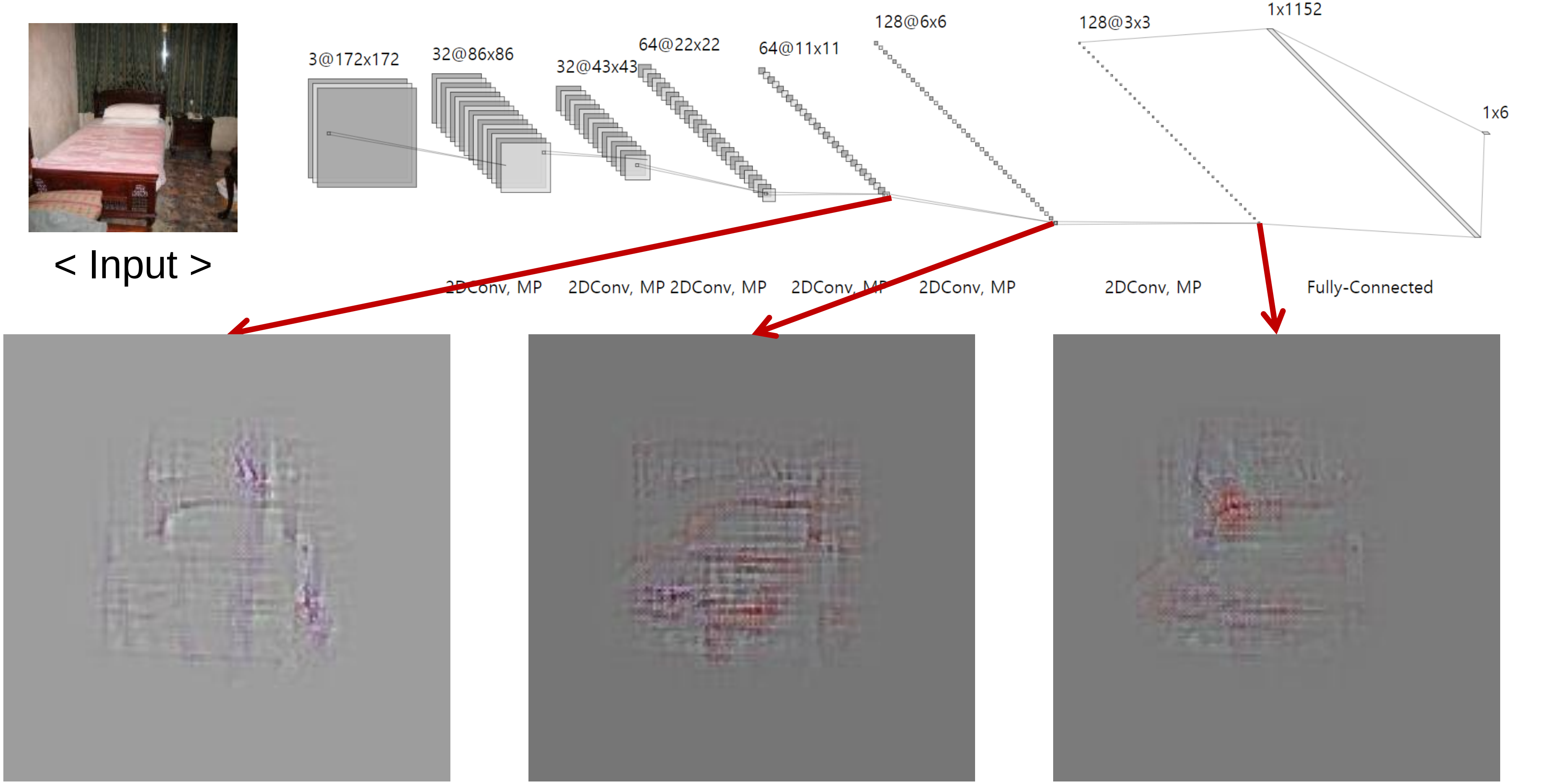
Visualization – Imagenet (Bed)

- In the front of layers, the model see the overall outline.



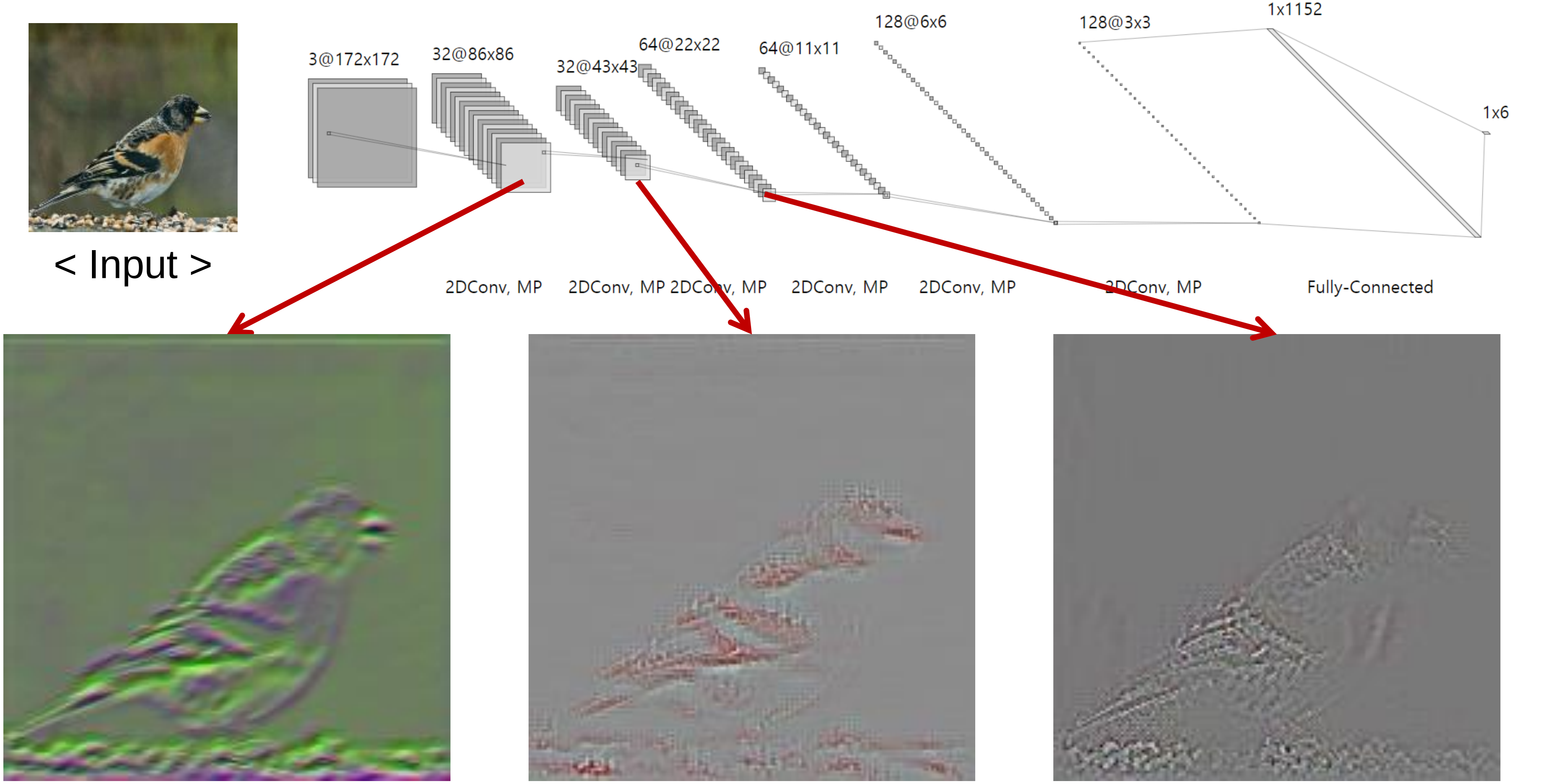
Visualization – Imagenet (Bed)

- In the backward of layers, the model see the detail object(ex.pillow).



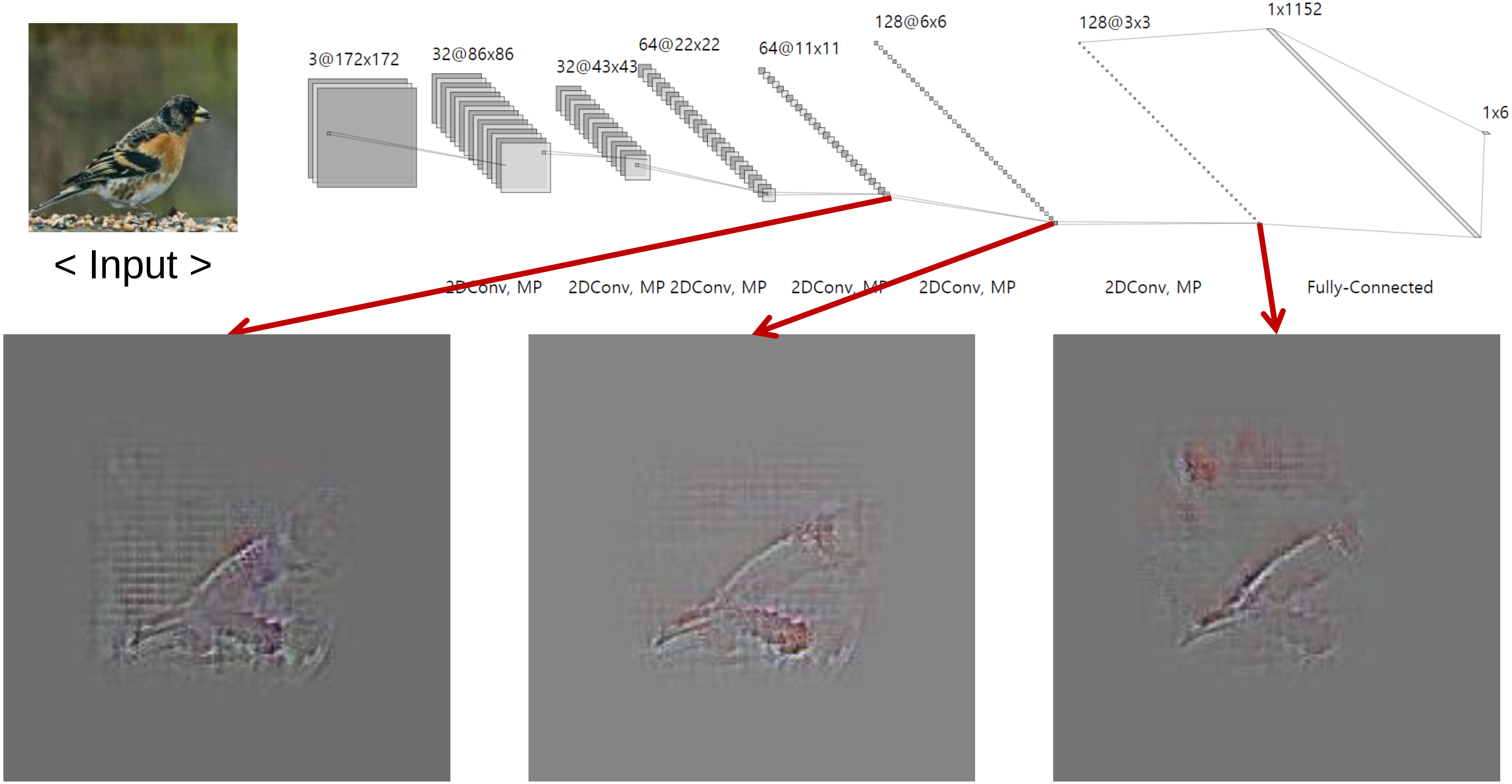
Visualization – Imagenet (Bird)

- In the front of layers, the model especially see the wing(ex. shape, pattern)



Visualization – Imagenet (Bird)

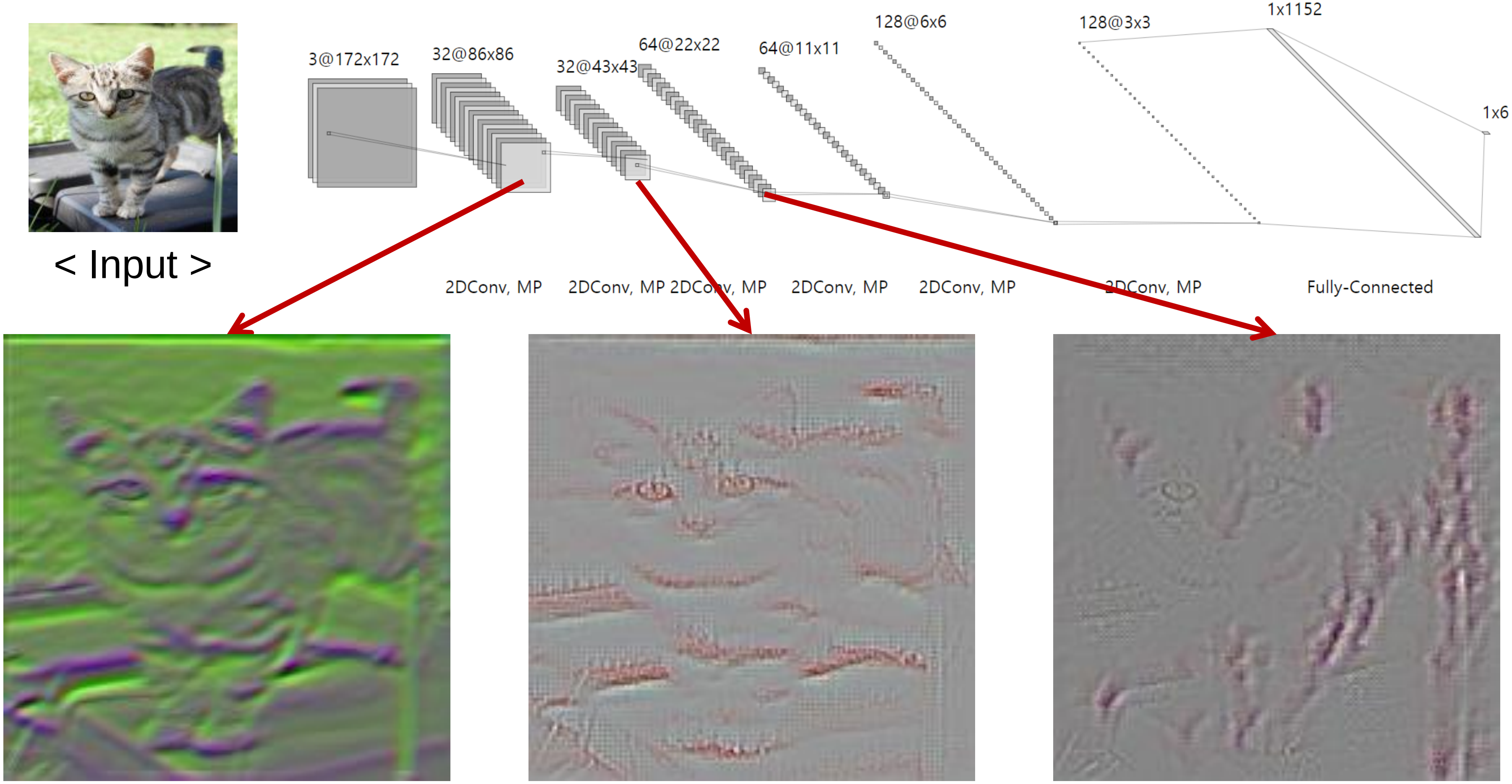
- In the backward of layers, the model also see the wing.
- The wing is important feature to classify whether it is bird or not.



< Input >

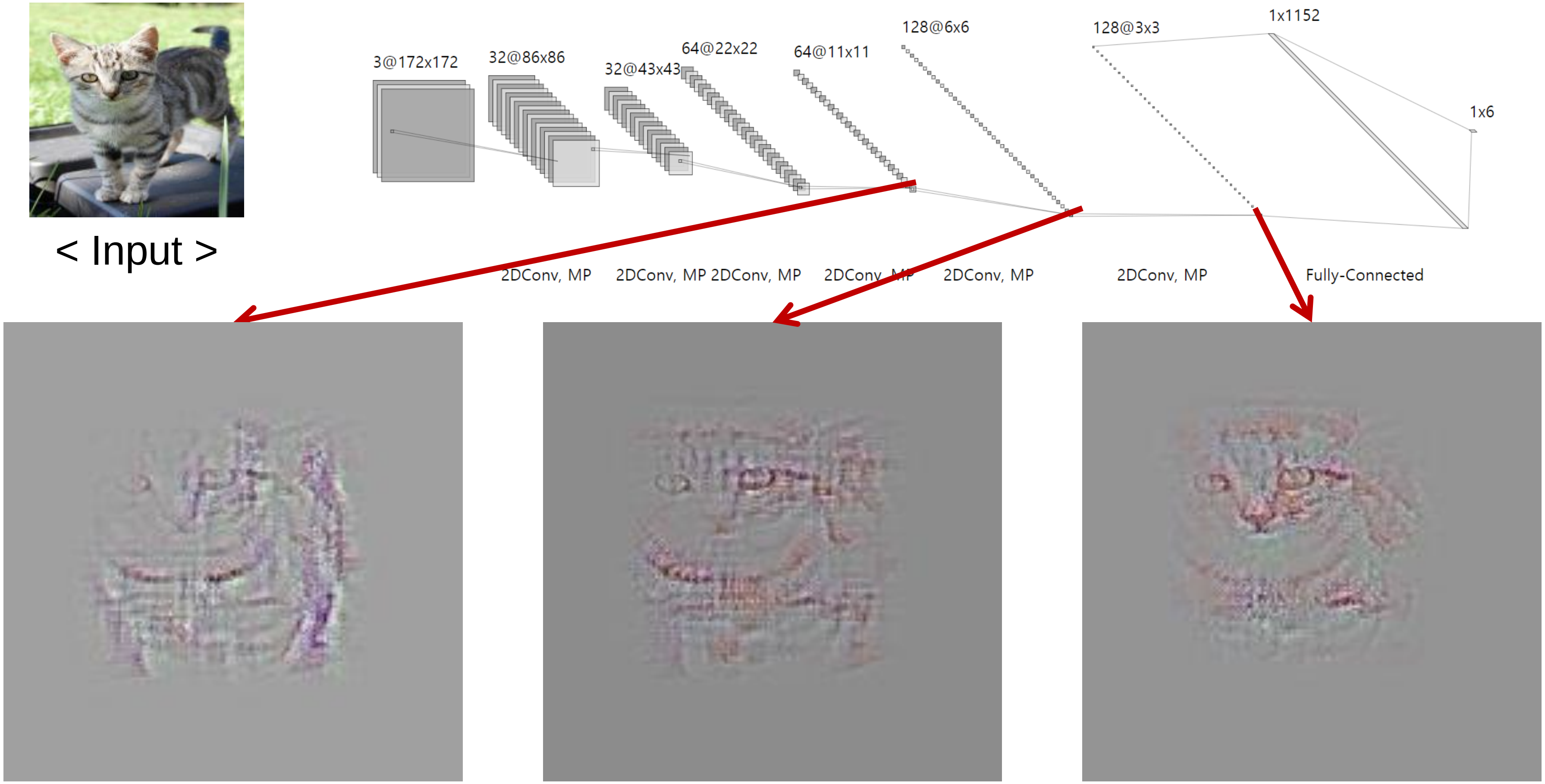
Visualization – Imagenet (Cat)

- In the second of layers, the model see the eyes and nose.
- In the third of layers, the model see the ears and tail.



Visualization – Imagenet (Cat)

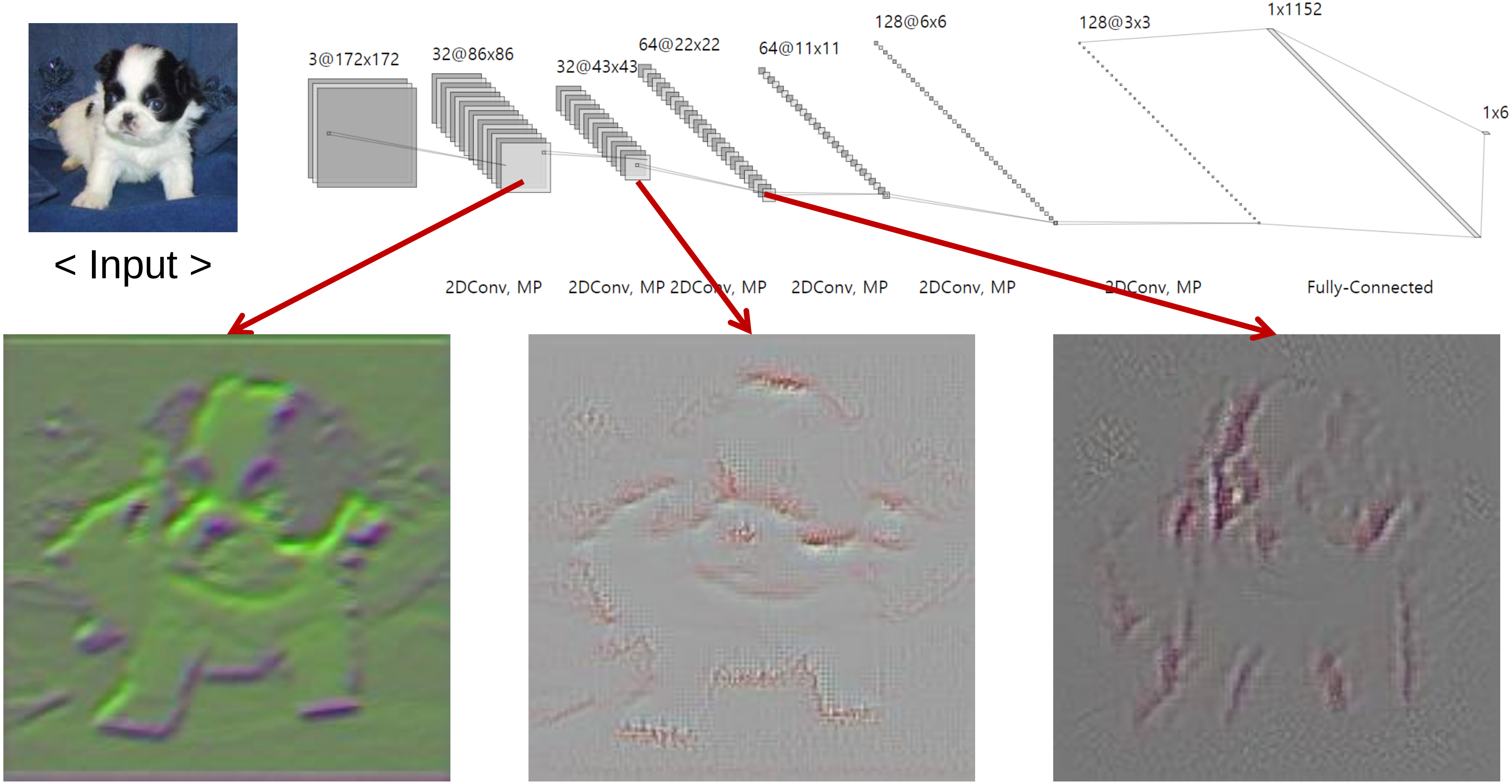
- In the backward of layers, the model see the overall face.
- Maybe this model see the nose to classify whether it is animal or not. (remind that this model confuse cat and dog)



< Input >

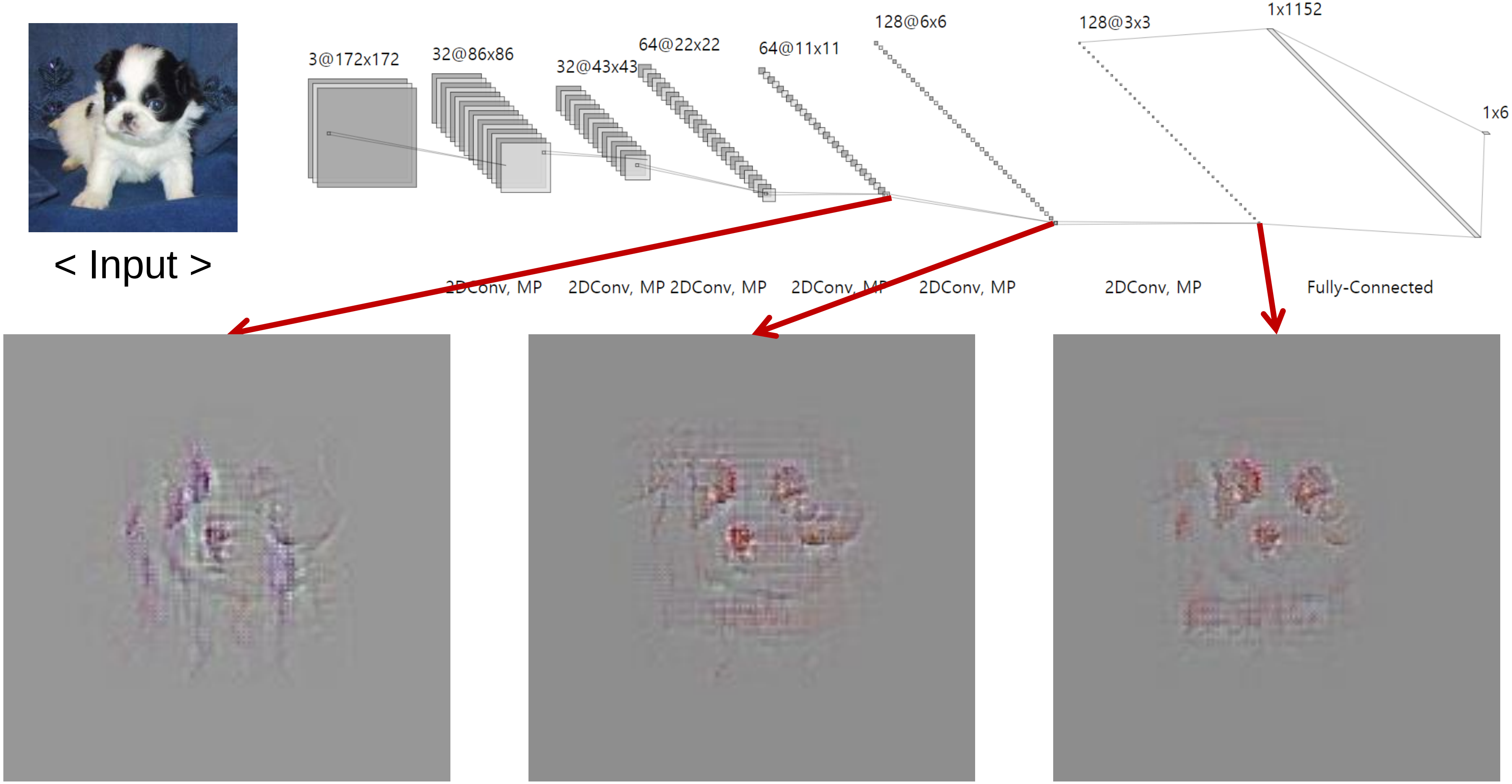
Visualization – Imagenet (Dog)

- In the second of layers, the model see the overall outline.
- In the third of layers, the model see the eyes and nose.



Visualization – Imagenet (Dog)

- In the backward of layers, the model see the overall face. (eyes, nose)
- And… the nose of cat and dog are very similar… I think…

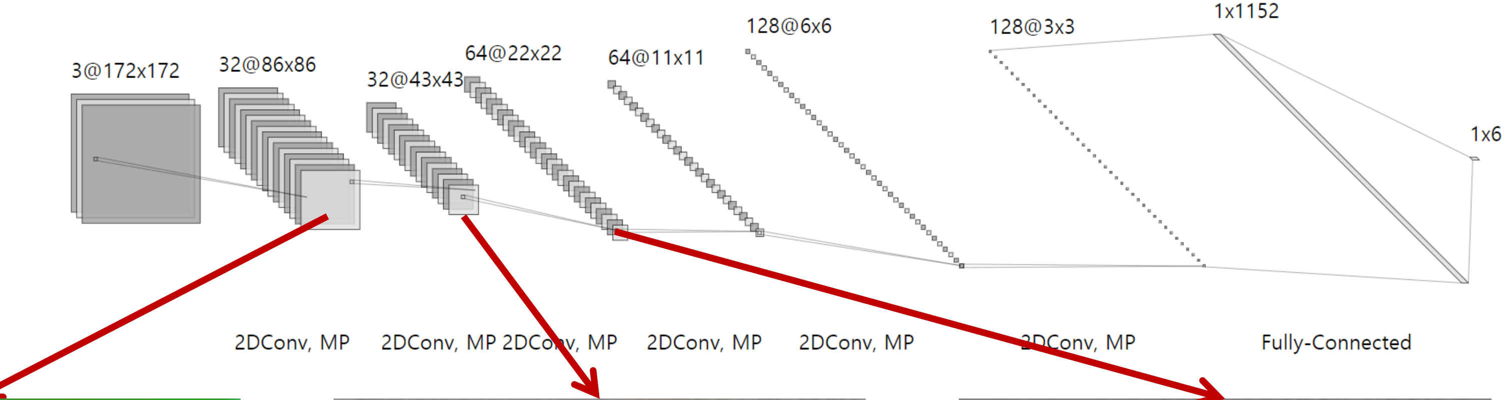


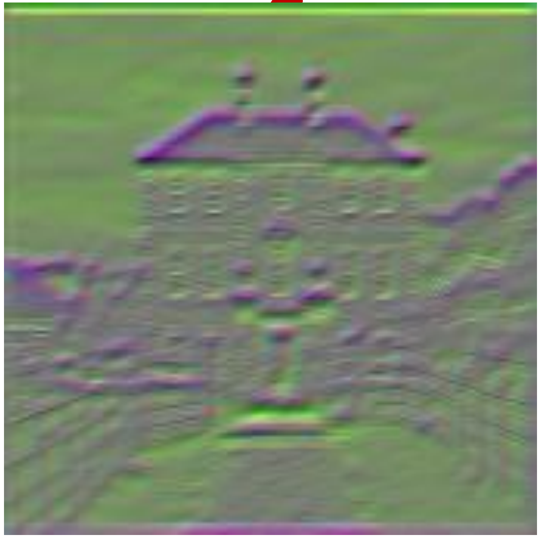
< Input >

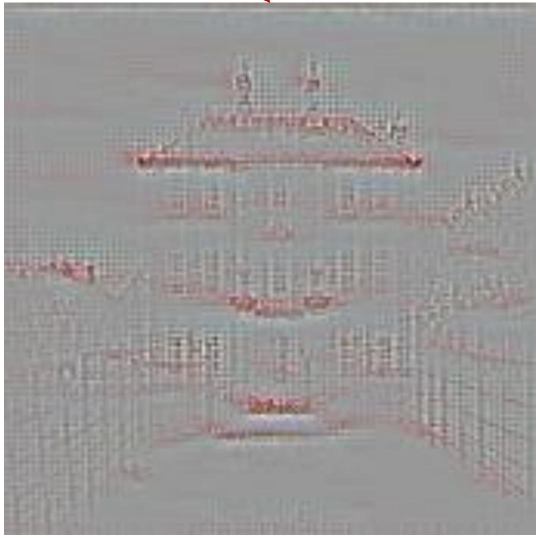
Visualization – Imagenet (House)

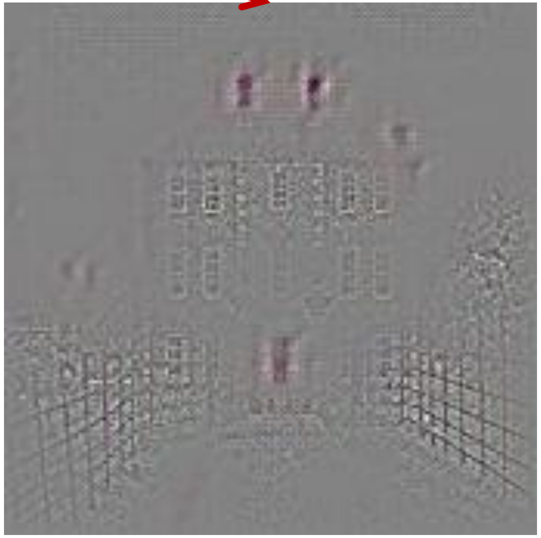
- In the front of layers, the model see the object of house (proof, window, fence, door)

< Input >





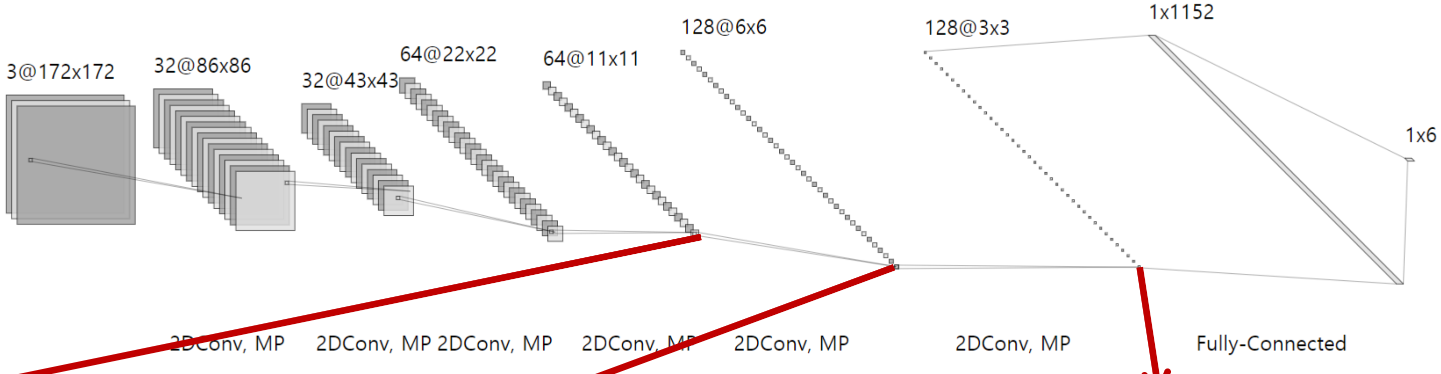


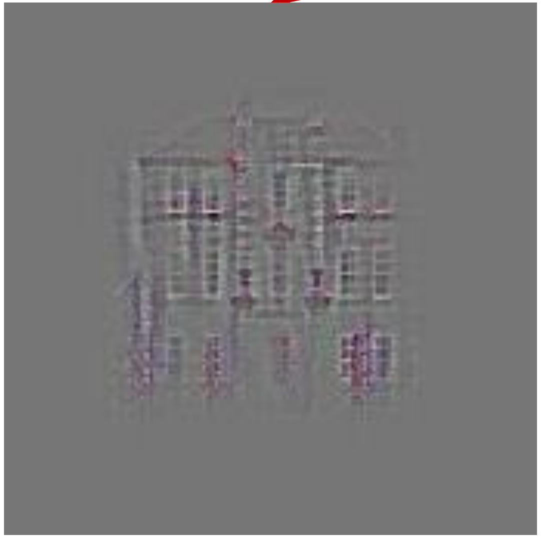


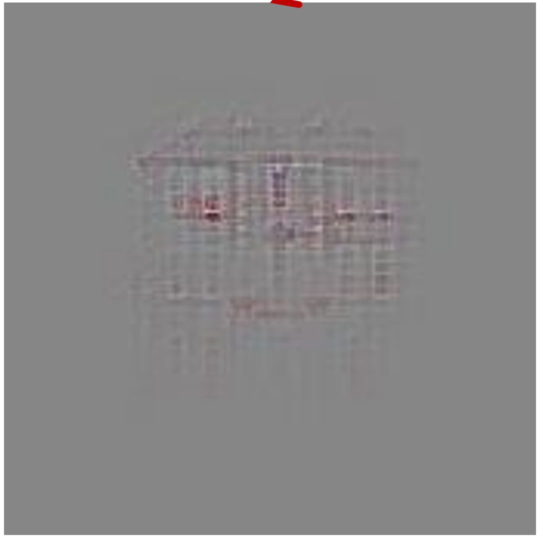
Visualization – Imagenet (House)

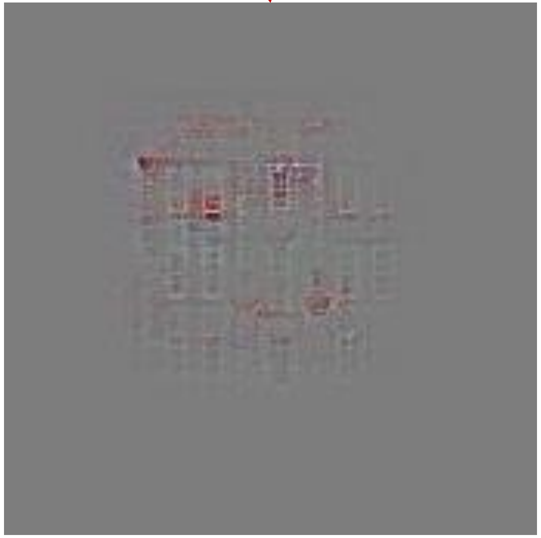
- In the backward layers, the model especially see the windows.
- The window is good feature to classify whether it is a house or not

< Input >



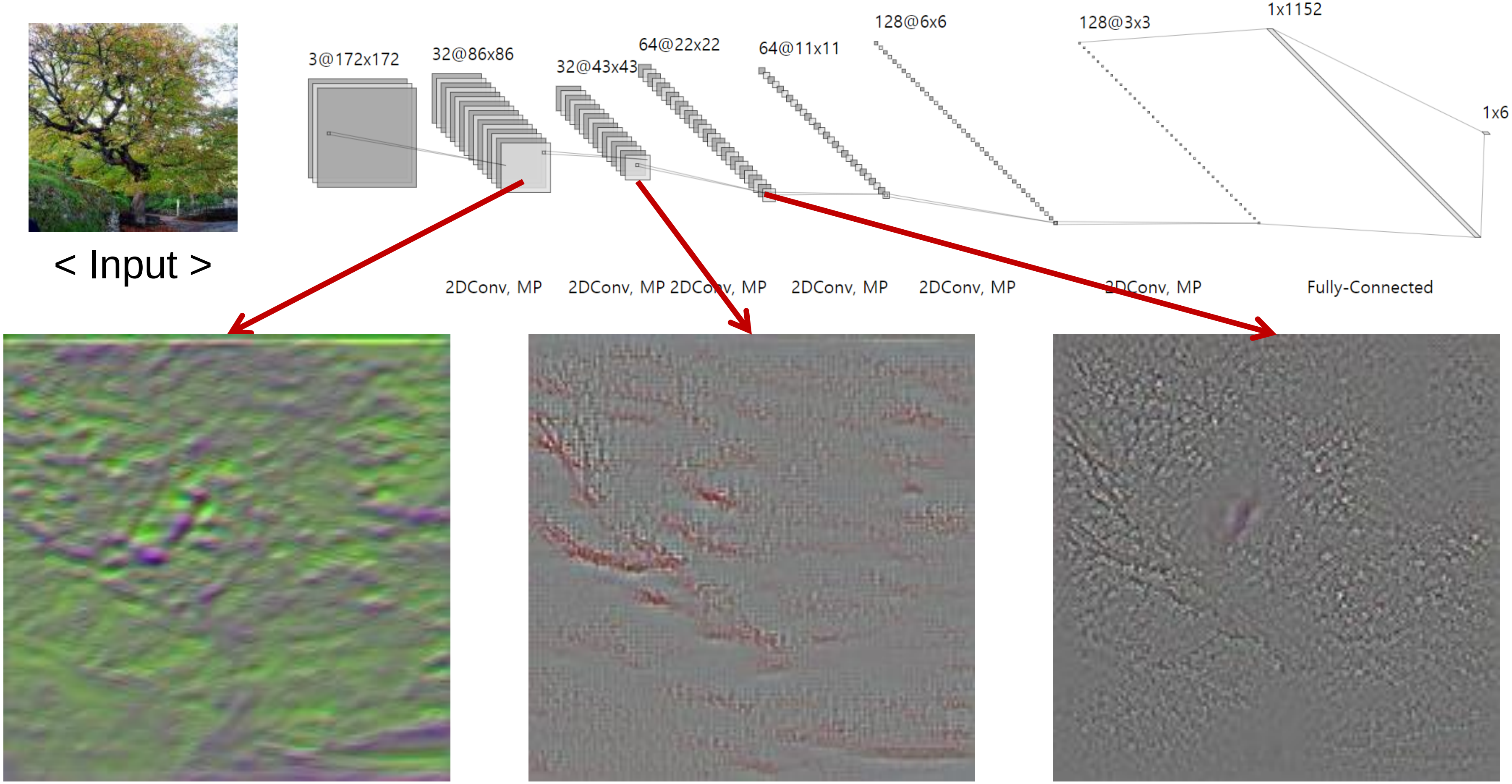






Visualization – Imagenet (Tree)

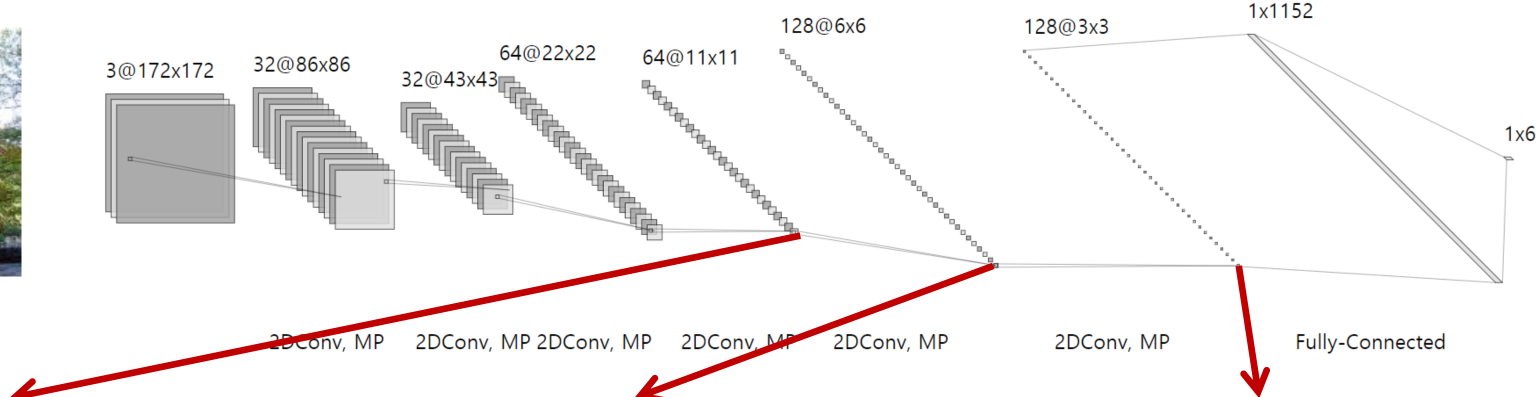
- In the second of layers, the model see the overall branch.
- In the third of layers, the model see the overall leaves.

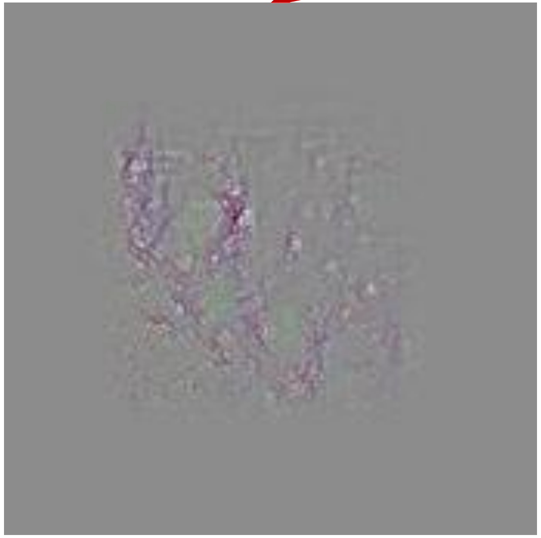


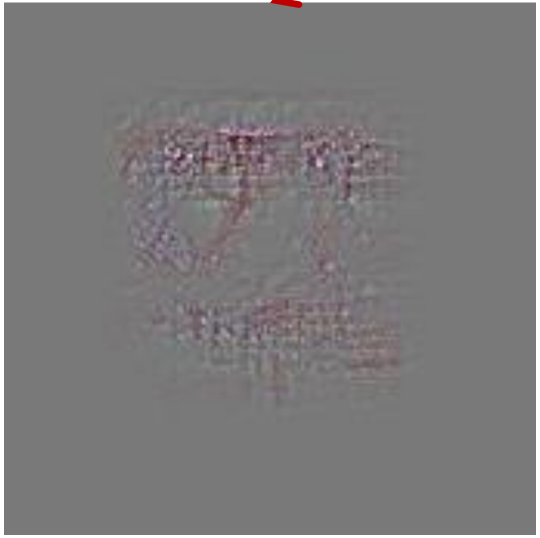
Visualization – Imagenet (Tree)

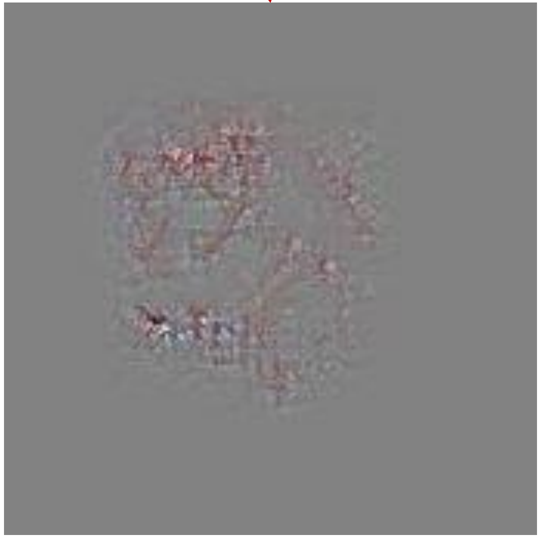
- In the backward of layers, the model see the branch and leaves both.

< Input >









Any Question?

Thank you