Lecture 30

Inside Our Computer Vision Model

Mohammad Sabik Irbaz
Data Scientist, Leadbook Pte. Ltd.
Former Lead ML Engineer, Omdena & Pioneer Alpha
sabikirbaz@iut-dhaka.edu

Dokkho Data Science Career Program
By MasterCourse

CODE Walkthrough

```
[ ]  1 !pip install -Uqq fastai
```

```
 1 from fastai.vision.all import *
 2 path = untar_data(URLs.PETS)/'images'
 3
 4 def is_cat(x): return x[0].isupper()
 5 dls = ImageDataLoaders.from_name_func(
 6     path, get_image_files(path), valid_pct=0.2, seed=42,
 7     label_func=is_cat, item_tfms=Resize(224))
 8
 9 learn = vision_learner(dls, resnet34, metrics=error_rate)
10 learn.fine_tune(2)
```

```
1 img = PILImage.create("cat.png")
2 img.to_thumb(192)
```

```
[ ]  1 is_cat,_,probs = learn.predict(img)
     2 print(f"Is this a cat?: {is_cat}.")
     3 print(f"Probability it's a cat: {probs[1].item():.6f}")
```

```
Is this a cat?: True.
Probability it's a cat: 1.000000
```

```
[ ] 1 !pip install -Uqq fastai
```

```
1 from fastai.vision.all import *
2 path = untar_data(URLs.PETS)/'images'
3
4 def is_cat(x): return x[0].isupper()
5 dls = ImageDataLoaders.from_name_func(
6     path, get_image_files(path), valid_pct=0.2, seed=42,
7     label_func=is_cat, item_tfms=Resize(224))
8
9 learn = vision_learner(dls, resnet34, metrics=error_rate)
10 learn.fine_tune(2)
```

```
1 img = PILImage.create("cat.png")
2 img.to_thumb(192)
```

```
[ ] 1 is_cat,_,probs = learn.predict(img)
2 print(f"Is this a cat?: {is_cat}.")
3 print(f"Probability it's a cat: {probs[1].item():.6f}")
```

```
Is this a cat?: True.
Probability it's a cat: 1.000000
```

Downloading and Installing the latest version of fastai.

Fastai library installs a lot of other necessary libraries like NumPy, Pandas, Torch

```
[ ]  1 !pip install -Uqq fastai
```

```
1 from fastai.vision.all import *
2 path = untar_data(URLS.PETS)/'images'
3
4 def is_cat(x): return x[0].isupper()
5 dls = ImageDataLoaders.from_name_func(
6     path, get_image_files(path), valid_pct=0.2, seed=42,
7     label_func=is_cat, item_tfms=Resize(224))
8
9 learn = vision_learner(dls, resnet34, metrics=error_rate)
10 learn.fine_tune(2)
```

```
1 img = PILImage.create("cat.png")
2 img.to_thumb(192)
```

```
[ ]  1 is_cat,_,probs = learn.predict(img)
     2 print(f"Is this a cat?: {is_cat}.")
     3 print(f"Probability it's a cat: {probs[1].item():.6f}")
```

```
Is this a cat?: True.
Probability it's a cat: 1.000000
```

Fastai has a module specifically built for vision application named “fastai.vision”

It contains different functions and libraries necessary for various computer vision models and applications.

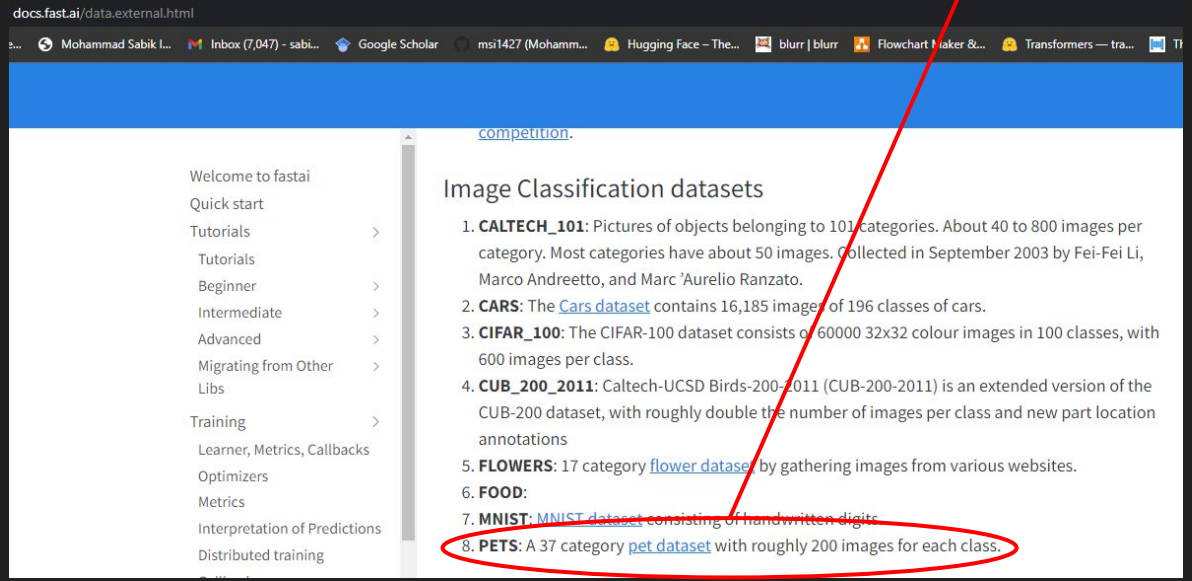
“ import * ” should be avoided when we put the code into production, but during the training and testing, it is convenient since we do not need to worry about importing necessary functions and/or libraries.

```
path = untar_data(URLs.PETS)/'images'
path
```

```
100.00% [811712512/811706944 01:25<00:00]
Path('/root/.fastai/data/oxford-iiit-pet/images')
```

Download the PETs data from fastai dataset.

Extracts the data under “images” folder



7393 images of different cats and dogs

```
1 len(path.ls())
```

```
7393
```

```
1 path.ls()[:5]
```

```
(#5) [Path('/root/.fastai/data/oxford-iiit-pet/images/Bombay_78.jpg'),Path('/root/.fastai/data/oxford-iiit-pet/images/chihuahua_166.jpg'),
pet/images/havanese_73.jpg'),Path('/root/.fastai/data/oxford-iiit-pet/images/Maine_Coon_77.jpg'),Path('/root/.fastai/data/oxford-iiit-pet/
```

```
1 img = PILImage.create(path.ls()[0])
2 img.to_thumb(192)
```

```
1 img = PILImage.create(path.ls()[1])
2 img.to_thumb(192)
```

```
1 len(path.ls())
```

```
7393
```

```
1 path.ls()[:5]
```

```
(#5) [Path('/root/.fastai/data/oxford-iiit-pet/images/Bombay_78.jpg'),Path('/root/.fastai/data/oxford-iiit-pet/images/chihuahua_166.jpg'),
pet/images/havanese_73.jpg'),Path('/root/.fastai/data/oxford-iiit-pet/images/Maine_Coon_77.jpg'),Path('/root/.fastai/data/oxford-iiit-pet/
```

```
1 img = PILImage.create(path.ls()[0])
2 img.to_thumb(192)
```

```
1 img = PILImage.create(path.ls()[1])
2 img.to_thumb(192)
```

7393 images of different cats and dogs

Cat Image name starts with Uppercase

Dog Image Name starts with Lowercase

Label function : returns 0 or 1

```
def is_cat(x): return x[0].isupper()
dls = ImageDataLoaders.from_name_func(
        path,
        get_image_files(path),
        valid_pct=0.2,
        seed=42,
        label_func=is_cat,
        item_tfms=Resize(224)
      )
```

ImageDataLoaders.from_name_func

source

```
ImageDataLoaders.from_name_func (path:str|Path, fnames:list,
                                 label_func:callable, valid_pct=0.2,
                                 seed=None, item_tfms=None,
                                 batch_tfms=None, img_cls=<class
                                 'fastai.vision.core.PILImage'>,
                                 bs:int=64, val_bs:int=None,
                                 shuffle:bool=True, device=None)
```

Create from the name attrs of `fnames` in `path`s with `label_func`

	Type	Default	Details
path	str | Path		Set the default path to a directory that a `Learner` can use to save files like models
fnames	list		A list of `os.Pathlike`'s to individual image files
label_func	callable		A function that receives a string (the file name) and outputs a label
valid_pct	float	0.2	
seed	NoneType	None	
item_tfms	NoneType	None	
batch_tfms	NoneType	None	
img_cls	BypassNewMeta	PILImage	
bs	int	64	Size of batch
val_bs	int	None	Size of batch for validation `DataLoader`
shuffle	bool	True	Whether to shuffle data
device	NoneType	None	Device to put `DataLoaders`
Returns	**DataLoaders**		

URL: https://docs.fast.ai/vision.data.html#imagedataloaders.from_name_func

```python
def is_cat(x): return x[0].isupper()
dls = ImageDataLoaders.from_name_func(
        path,
        get_image_files(path),
        valid_pct=0.2,
        seed=42,
        label_func=is_cat,
        item_tfms=Resize(224)
      )
```

ImageDataLoaders.from_name_func

source

```
ImageDataLoaders.from_name_func (path:str|Path, fnames:list,
                                 label_func:callable, valid_pct=0.2,
                                 seed=None, item_tfms=None,
                                 batch_tfms=None, img_cls=<class
                                 'fastai.vision.core.PILImage'>,
                                 bs:int=64, val_bs:int=None,
                                 shuffle:bool=True, device=None)
```

Create from the name attrs of `fnames` in `path`s with `label_func`

	Type	Default	Details
path	str | Path		Set the default path to a directory that a `Learner` can use to save files like models
fnames	list		A list of `os.Pathlike`'s to individual image files
label_func	callable		A function that receives a string (the file name) and outputs a label
valid_pct	float	0.2	
seed	NoneType	None	
item_tfms	NoneType	None	
batch_tfms	NoneType	None	
img_cls	BypassNewMeta	PILImage	
bs	int	64	Size of batch
val_bs	int	None	Size of batch for validation `DataLoader`
shuffle	bool	True	Whether to shuffle data
device	NoneType	None	Device to put `DataLoaders`
Returns	**DataLoaders**		

```
1 learn = vision_learner(dls, resnet34, metrics=error_rate)
```

URL: https://docs.fast.ai/vision.learner.html#vision_learner

vision_learner

```
vision_learner (dls, arch, normalize=True, n_out=None, pretrained=True,
                loss_func=None, opt_func=<function Adam>, lr=0.001,
                splitter=None, cbs=None, metrics=None, path=None,
                model_dir='models', wd=None, wd_bn_bias=False,
                train_bn=True, moms=(0.95, 0.85, 0.95), cut=None,
                init=<function kaiming_normal_>, custom_head=None,
                concat_pool=True, pool=True, lin_ftrs=None, ps=0.5,
                first_bn=True, bn_final=False, lin_first=False,
                y_range=None, n_in=3)
```

Build a vision learner from `dls` and `arch`

```
1 learn = vision_learner(dls, resnet34, metrics=error_rate)
```

URL: https://fastai1.fast.ai/vision.models.html

URL: https://docs.fast.ai/vision.learner.html#vision_learner

vision_learner

```
vision_learner (dls, arch, normalize=True, n_out=None, pretrained=True,
                loss_func=None, opt_func=<function Adam>, lr=0.001,
                splitter=None, cbs=None, metrics=None, path=None,
                model_dir='models', wd=None, wd_bn_bias=False,
                train_bn=True, moms=(0.95, 0.85, 0.95), cut=None,
                init=<function kaiming_normal_>, custom_head=None,
                concat_pool=True, pool=True, lin_ftrs=None, ps=0.5,
                first_bn=True, bn_final=False, lin_first=False,
                y_range=None, n_in=3)
```

Build a vision learner from `dls` and `arch`

```
1 learn = vision_learner(dls, resnet34, metrics=error_rate)
```

Computer Vision models zoo

The fastai library includes several pretrained models from torchvision, namely:

- resnet18, resnet34, resnet50, resnet101, resnet152
- squeezenet1_0, squeezenet1_1
- densenet121, densenet169, densenet201, densenet161
- vgg16_bn, vgg19_bn
- alexnet

URL: https://fastai1.fast.ai/vision.models.html

URL: https://docs.fast.ai/metrics.html#error_rate

URL: https://docs.fast.ai/vision.learner.html#vision_learner

vision_learner

```
vision_learner (dls, arch, normalize=True, n_out=None, pretrained=True,
                loss_func=None, opt_func=<function Adam>, lr=0.001,
                splitter=None, cbs=None, metrics=None, path=None,
                model_dir='models', wd=None, wd_bn_bias=False,
                train_bn=True, moms=(0.95, 0.85, 0.95), cut=None,
                init=<function kaiming_normal_>, custom_head=None,
                concat_pool=True, pool=True, lin_ftrs=None, ps=0.5,
                first_bn=True, bn_final=False, lin_first=False,
                y_range=None, n_in=3)
```

Build a vision learner from `dls` and `arch`

```
1 learn = vision_learner(dls, resnet34, metrics=error_rate)
```

Computer Vision models zoo

The fastai library includes several pretrained models from torchvision, namely:

- resnet18, resnet34, resnet50, resnet101, resnet152
- squeezenet1_0, squeezenet1_1
- densenet121, densenet169, densenet201, densenet161
- vgg16_bn, vgg19_bn
- alexnet

error_rate

```
error_rate (inp, targ, axis=-1)
```

1 - accuracy

```
x = torch.randn(4,5)
y = x.argmax(dim=1)
test_eq(error_rate(x,y), 0)
y1 = change_targ(y, 2, 5)
test_eq(error_rate(x,y1), 0.5)
test_eq(error_rate(x.unsqueeze(1).expand(4,2,5), torch.stack([y,y1], dim=1)), 0.25)
```

```
1 learn.fine_tune(2)
```

epoch	train_loss	valid_loss	error_rate	time
0	0.158167	0.033317	0.009472	00:53
epoch	train_loss	valid_loss	error_rate	time
0	0.058523	0.038277	0.007442	00:51
1	0.029566	0.018829	0.006089	00:51

1. Adds custom layers
2. Freezes the pretrained layers
3. Train for one epoch
4. Unfreeze the pretrained layers
5. Train for “n” epochs

Inference

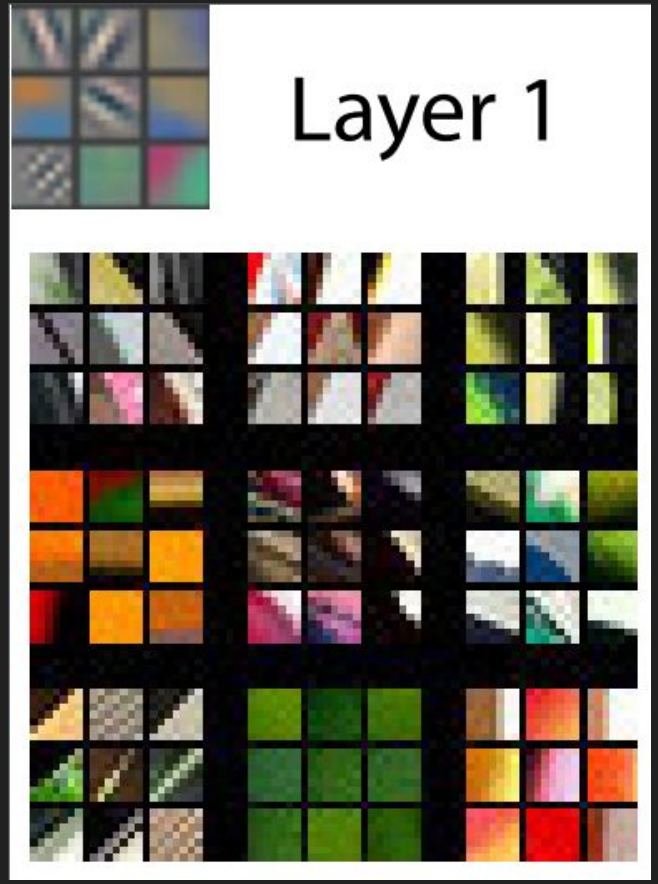
```
1 learn.predict(img)
```

```
('True', TensorBase(1), TensorBase([3.8827e-12, 1.0000e+00]))
```

```
1 is_cat,_,probs = learn.predict(img)
2 print(f"Is this a cat?: {is_cat}.")
3 print(f"Probability it's a cat: {probs[1].item():.6f}")
```

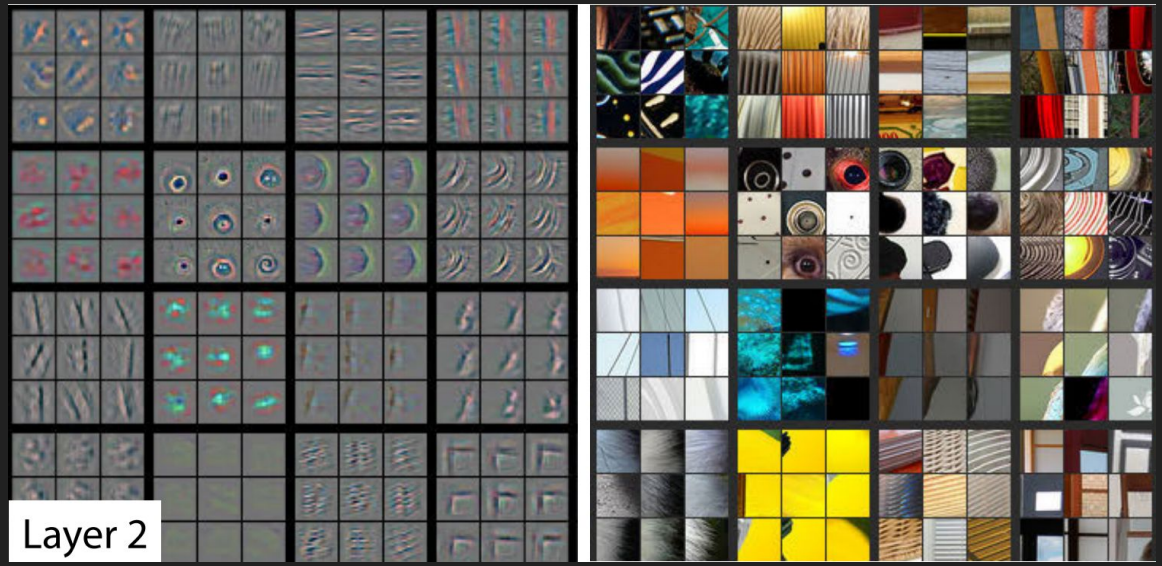
```
Is this a cat?: True.
Probability it's a cat: 1.000000
```

What Our Model Learns



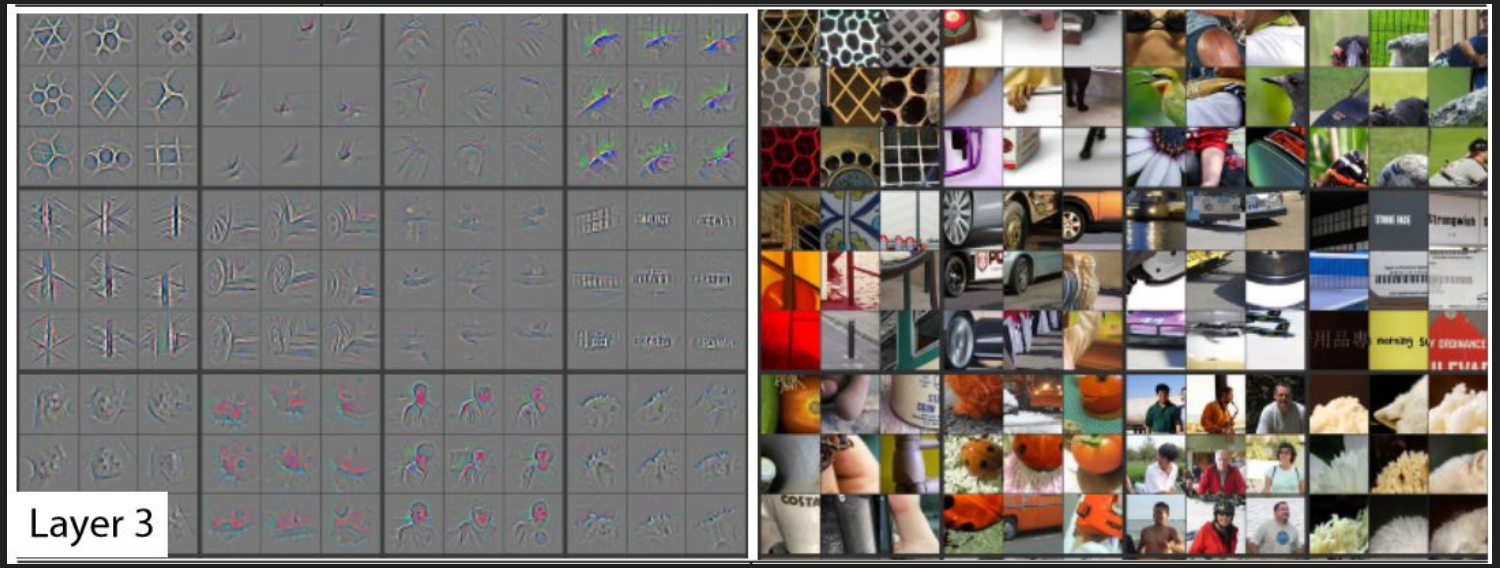
From “Visualizing and Understanding Convolutional Networks” Paper

What Our Model Learns



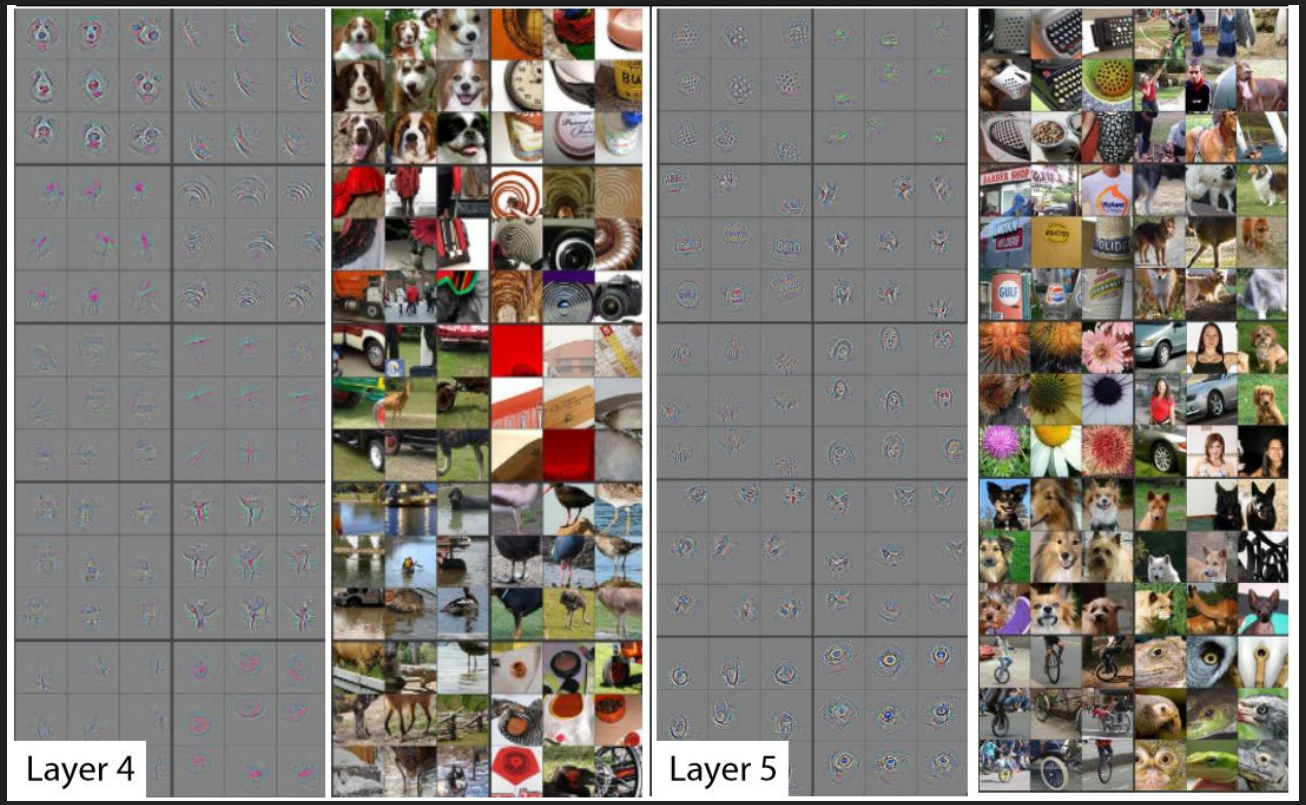
From “Visualizing and Understanding Convolutional Networks” Paper

What Our Model Learns



From “Visualizing and Understanding Convolutional Networks” Paper

What Our Model Learns



From “Visualizing and Understanding Convolutional Networks” Paper