

```
In [1]: import numpy as np
import tensorflow as tf
import cv2
import matplotlib.pyplot as plt
%matplotlib inline
from keras.models import Sequential
from keras.layers import Input, Concatenate
from keras.layers import Conv2D, Conv2DTranspose, Activation, MaxPooling2D, UpSampling2D
from keras.layers.advanced_activations import PReLU, LeakyReLU
from keras.models import Model
from keras.models import load_model
from tensorflow.keras.optimizers import Adam
from keras import losses
```

2021-11-28 23:12:19.990611: W tensorflow/stream\_executor/platform/default/dso\_loader.cc:64] Could not load dynamic library 'libcudart.so.11.0'; dlerror: libcudart.so.11.0: cannot open shared object file: No such file or directory  
2021-11-28 23:12:19.990659: I tensorflow/stream\_executor/cuda/cudart\_stub.cc:29] Ignore above cudart dlerror if you do not have a GPU set up on your machine.

```
In [2]: rain_images = []
"""
for i in glob.glob("/content/drive/My Drive/AI Project/Dataset/Train/data/*"):
    rain_images.append(cv2.imread(i))

for i in glob.glob("/content/drive/My Drive/AI Project/Dataset/Train2/data/*"):
    rain_images.append(cv2.imread(i))
"""

for i in range(861):
    file = "Dataset/Train/data/" + str(i) + "_rain.png"
    rain_images.append(cv2.imread(file))

for i in range(249):
    file = "Dataset/Train2/data/" + str(i) + "_rain.jpg"
    rain_images.append(cv2.imread(file))
```

```
In [3]: derain_images = []
"""
for i in glob.glob("/content/drive/My Drive/AI Project/Dataset/Train/gt/*"):
    derain_images.append(cv2.imread(i))

for i in glob.glob("/content/drive/My Drive/AI Project/Dataset/Train2/gt/*"):
    derain_images.append(cv2.imread(i))
"""

for i in range(861):
    file = "Dataset/Train/gt/" + str(i) + "_clean.png"
    derain_images.append(cv2.imread(file))

for i in range(249):
    file = "Dataset/Train2/gt/" + str(i) + "_clean.jpg"
    derain_images.append(cv2.imread(file))
```

```
In [4]: rain_count = 0
derain_count = 0
```

```
rain_final = []
derain_final = []
for i in rain_images:
    rain_count +=1
    x = cv2.resize(i,(512,512))
    x = x/255
    rain_final.append(x)

for i in derain_images:
    derain_count +=1
    x = cv2.resize(i,(512,512))
    x = x/255
    derain_final.append(x)
```

In [5]:

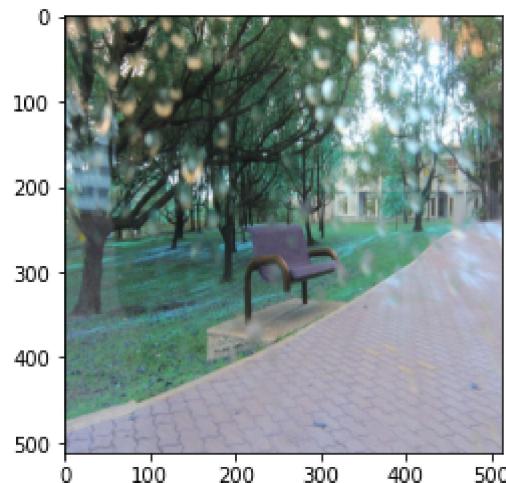
```
print(rain_count)
print(derain_count)
```

```
1110
1110
```

In [6]:

```
plt.imshow(rain_final[20])
```

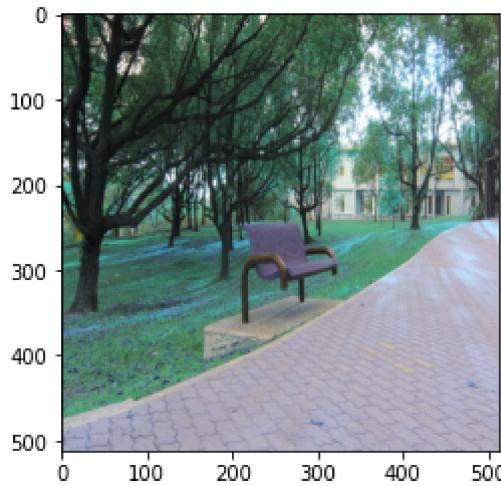
Out[6]:



In [7]:

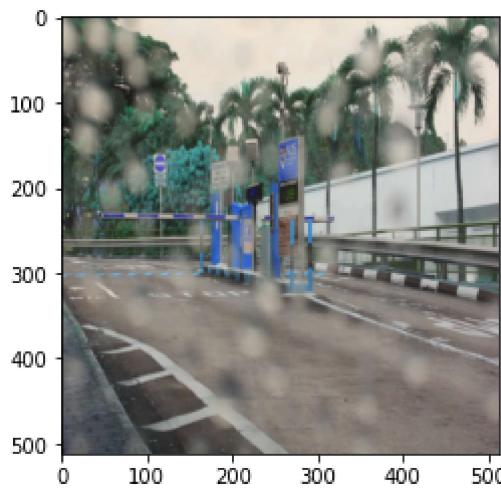
```
plt.imshow(derain_final[20])
```

Out[7]:



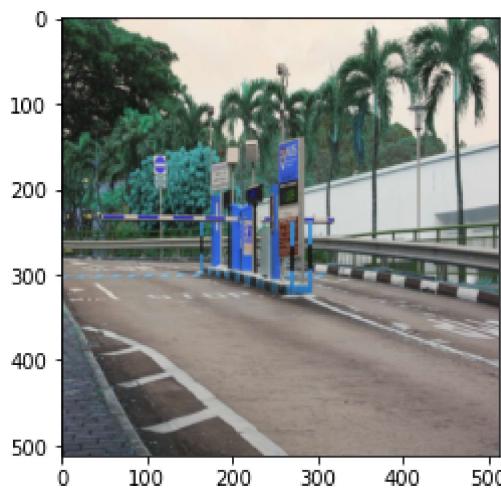
In [8]: `plt.imshow(rain_final[100])`

Out[8]: <matplotlib.image.AxesImage at 0x7fe6800afe20>



In [9]: `plt.imshow(derain_final[100])`

Out[9]: <matplotlib.image.AxesImage at 0x7fe332d3d250>



In [10]:

```
rain_final = np.asarray(rain_final)
rain_final.shape
```

Out[10]: (1110, 512, 512, 3)

In [11]:

```
derain_final = np.asarray(derain_final)
derain_final.shape
```

Out[11]: (1110, 512, 512, 3)

In [12]:

```
input_img = Input(shape=(512, 512, 3))

x = Conv2D(8, (4, 4), activation='relu', padding='same')(input_img)
x = MaxPooling2D((2, 2), padding='same')(x)

x = Conv2D(16, (4, 4), activation='relu', padding='same')(x)
x = MaxPooling2D((2, 2), padding='same')(x)

x = Conv2D(32, (5, 5), activation='relu', padding='same')(x)
x = MaxPooling2D((2, 2), padding='same')(x)

encoded = Conv2D(64, (3, 3), activation='relu', padding='same')(x)

# at this point the representation is (64, 64, 64)

x = Conv2DTranspose(32, (3, 3), activation='relu', padding='same')(encoded)
x = UpSampling2D((2, 2))(x)
x = Conv2D(16, (5, 5), activation='relu', padding='same')(x)
x = UpSampling2D((2, 2))(x)
x = Conv2D(8, (5, 5), activation='relu', padding='same')(x)
x = UpSampling2D((2, 2))(x)
decoded = Conv2D(3, (3, 3), activation='sigmoid', padding = 'same')(x)

autoencoder = Model(input_img, decoded)
autoencoder.summary()
autoencoder.compile(optimizer='adam', loss=losses.mean_squared_error, metrics = ['accu
```

2021-11-28 23:13:23.269801: W tensorflow/stream\_executor/platform/default/dso\_loader.cc:64] Could not load dynamic library 'libcuda.so.1'; dlerror: libcuda.so.1: cannot open shared object file: No such file or directory; LD\_LIBRARY\_PATH: /home/siddhaj3/.local/lib/python3.8/site-packages/cv2/.../lib64:

2021-11-28 23:13:23.269854: W tensorflow/stream\_executor/cuda/cuda\_driver.cc:269] failed call to cuInit: UNKNOWN ERROR (303)

2021-11-28 23:13:23.269918: I tensorflow/stream\_executor/cuda/cuda\_diagnostics.cc:156] kernel driver does not appear to be running on this host (ai-model-instance): /proc/drive r/nvidia/version does not exist

2021-11-28 23:13:23.270291: I tensorflow/core/platform/cpu\_feature\_guard.cc:151] This TensorFlow binary is optimized with oneAPI Deep Neural Network Library (oneDNN) to use the following CPU instructions in performance-critical operations: AVX2 AVX512F FMA To enable them in other operations, rebuild TensorFlow with the appropriate compiler flags.

Model: "model"

Layer (type)	Output Shape	Param #
input_1 (InputLayer)	[(None, 512, 512, 3)]	0

conv2d (Conv2D)	(None, 512, 512, 8)	392
max_pooling2d (MaxPooling2D )	(None, 256, 256, 8)	0
conv2d_1 (Conv2D)	(None, 256, 256, 16)	2064
max_pooling2d_1 (MaxPooling 2D)	(None, 128, 128, 16)	0
conv2d_2 (Conv2D)	(None, 128, 128, 32)	12832
max_pooling2d_2 (MaxPooling 2D)	(None, 64, 64, 32)	0
conv2d_3 (Conv2D)	(None, 64, 64, 64)	18496
conv2d_transpose (Conv2DTra nspose)	(None, 64, 64, 32)	18464
up_sampling2d (UpSampling2D )	(None, 128, 128, 32)	0
conv2d_4 (Conv2D)	(None, 128, 128, 16)	12816
up_sampling2d_1 (UpSampling 2D)	(None, 256, 256, 16)	0
conv2d_5 (Conv2D)	(None, 256, 256, 8)	3208
up_sampling2d_2 (UpSampling 2D)	(None, 512, 512, 8)	0
conv2d_6 (Conv2D)	(None, 512, 512, 3)	219

---

Total params: 68,491  
 Trainable params: 68,491  
 Non-trainable params: 0

---

In [13]:

```
history = Model.fit(autoencoder, rain_final, derain_final, validation_split = 0.2, epoch
```

```
2021-11-28 23:13:25.024890: W tensorflow/core/framework/cpu_allocator_impl.cc:82] Allocation of 2793406464 exceeds 10% of free system memory.
2021-11-28 23:13:28.618167: W tensorflow/core/framework/cpu_allocator_impl.cc:82] Allocation of 2793406464 exceeds 10% of free system memory.
Epoch 1/100
56/56 [=====] - 235s 4s/step - loss: 0.0421 - accuracy: 0.3604
- val_loss: 0.0212 - val_accuracy: 0.3898
Epoch 2/100
56/56 [=====] - 236s 4s/step - loss: 0.0174 - accuracy: 0.4769
- val_loss: 0.0173 - val_accuracy: 0.5667
Epoch 3/100
56/56 [=====] - 234s 4s/step - loss: 0.0168 - accuracy: 0.5380
- val_loss: 0.0147 - val_accuracy: 0.5514
Epoch 4/100
56/56 [=====] - 231s 4s/step - loss: 0.0147 - accuracy: 0.6021
```

```
- val_loss: 0.0167 - val_accuracy: 0.5601
Epoch 5/100
56/56 [=====] - 226s 4s/step - loss: 0.0151 - accuracy: 0.5954
- val_loss: 0.0138 - val_accuracy: 0.5674
Epoch 6/100
56/56 [=====] - 224s 4s/step - loss: 0.0138 - accuracy: 0.6118
- val_loss: 0.0137 - val_accuracy: 0.6046
Epoch 7/100
56/56 [=====] - 224s 4s/step - loss: 0.0136 - accuracy: 0.6481
- val_loss: 0.0125 - val_accuracy: 0.6044
Epoch 8/100
56/56 [=====] - 224s 4s/step - loss: 0.0133 - accuracy: 0.6380
- val_loss: 0.0128 - val_accuracy: 0.5580
Epoch 9/100
56/56 [=====] - 224s 4s/step - loss: 0.0124 - accuracy: 0.6595
- val_loss: 0.0113 - val_accuracy: 0.5840
Epoch 10/100
56/56 [=====] - 227s 4s/step - loss: 0.0122 - accuracy: 0.6644
- val_loss: 0.0125 - val_accuracy: 0.6771
Epoch 11/100
56/56 [=====] - 224s 4s/step - loss: 0.0117 - accuracy: 0.6610
- val_loss: 0.0108 - val_accuracy: 0.5978
Epoch 12/100
56/56 [=====] - 224s 4s/step - loss: 0.0115 - accuracy: 0.6642
- val_loss: 0.0107 - val_accuracy: 0.6129
Epoch 13/100
56/56 [=====] - 229s 4s/step - loss: 0.0116 - accuracy: 0.6604
- val_loss: 0.0107 - val_accuracy: 0.6625
Epoch 14/100
56/56 [=====] - 226s 4s/step - loss: 0.0115 - accuracy: 0.6576
- val_loss: 0.0109 - val_accuracy: 0.6497
Epoch 15/100
56/56 [=====] - 226s 4s/step - loss: 0.0115 - accuracy: 0.6471
- val_loss: 0.0111 - val_accuracy: 0.6043
Epoch 16/100
56/56 [=====] - 226s 4s/step - loss: 0.0110 - accuracy: 0.6693
- val_loss: 0.0102 - val_accuracy: 0.6450
Epoch 17/100
56/56 [=====] - 225s 4s/step - loss: 0.0106 - accuracy: 0.6917
- val_loss: 0.0100 - val_accuracy: 0.6708
Epoch 18/100
56/56 [=====] - 226s 4s/step - loss: 0.0106 - accuracy: 0.6803
- val_loss: 0.0101 - val_accuracy: 0.6907
Epoch 19/100
56/56 [=====] - 226s 4s/step - loss: 0.0106 - accuracy: 0.6802
- val_loss: 0.0101 - val_accuracy: 0.6301
Epoch 20/100
56/56 [=====] - 224s 4s/step - loss: 0.0106 - accuracy: 0.6657
- val_loss: 0.0098 - val_accuracy: 0.6656
Epoch 21/100
56/56 [=====] - 229s 4s/step - loss: 0.0102 - accuracy: 0.6809
- val_loss: 0.0100 - val_accuracy: 0.6270
Epoch 22/100
56/56 [=====] - 223s 4s/step - loss: 0.0099 - accuracy: 0.6931
- val_loss: 0.0096 - val_accuracy: 0.6865
Epoch 23/100
56/56 [=====] - 224s 4s/step - loss: 0.0099 - accuracy: 0.6940
- val_loss: 0.0101 - val_accuracy: 0.6381
Epoch 24/100
56/56 [=====] - 225s 4s/step - loss: 0.0100 - accuracy: 0.6967
```

```
- val_loss: 0.0095 - val_accuracy: 0.6132
Epoch 25/100
56/56 [=====] - 224s 4s/step - loss: 0.0099 - accuracy: 0.6836
- val_loss: 0.0102 - val_accuracy: 0.6508
Epoch 26/100
56/56 [=====] - 226s 4s/step - loss: 0.0097 - accuracy: 0.6885
- val_loss: 0.0093 - val_accuracy: 0.6789
Epoch 27/100
56/56 [=====] - 227s 4s/step - loss: 0.0096 - accuracy: 0.6991
- val_loss: 0.0093 - val_accuracy: 0.6425
Epoch 28/100
56/56 [=====] - 224s 4s/step - loss: 0.0096 - accuracy: 0.6782
- val_loss: 0.0098 - val_accuracy: 0.6791
Epoch 29/100
56/56 [=====] - 227s 4s/step - loss: 0.0099 - accuracy: 0.6883
- val_loss: 0.0095 - val_accuracy: 0.6145
Epoch 30/100
56/56 [=====] - 227s 4s/step - loss: 0.0094 - accuracy: 0.6887
- val_loss: 0.0090 - val_accuracy: 0.6869
Epoch 31/100
56/56 [=====] - 223s 4s/step - loss: 0.0093 - accuracy: 0.6889
- val_loss: 0.0098 - val_accuracy: 0.6642
Epoch 32/100
56/56 [=====] - 224s 4s/step - loss: 0.0096 - accuracy: 0.6745
- val_loss: 0.0098 - val_accuracy: 0.6811
Epoch 33/100
56/56 [=====] - 225s 4s/step - loss: 0.0093 - accuracy: 0.6875
- val_loss: 0.0099 - val_accuracy: 0.6946
Epoch 34/100
56/56 [=====] - 223s 4s/step - loss: 0.0095 - accuracy: 0.6731
- val_loss: 0.0091 - val_accuracy: 0.6859
Epoch 35/100
56/56 [=====] - 223s 4s/step - loss: 0.0092 - accuracy: 0.6917
- val_loss: 0.0112 - val_accuracy: 0.6922
Epoch 36/100
56/56 [=====] - 225s 4s/step - loss: 0.0094 - accuracy: 0.6751
- val_loss: 0.0092 - val_accuracy: 0.6757
Epoch 37/100
56/56 [=====] - 223s 4s/step - loss: 0.0091 - accuracy: 0.6955
- val_loss: 0.0091 - val_accuracy: 0.6267
Epoch 38/100
56/56 [=====] - 223s 4s/step - loss: 0.0092 - accuracy: 0.6727
- val_loss: 0.0088 - val_accuracy: 0.6590
Epoch 39/100
56/56 [=====] - 223s 4s/step - loss: 0.0090 - accuracy: 0.7053
- val_loss: 0.0087 - val_accuracy: 0.6778
Epoch 40/100
56/56 [=====] - 224s 4s/step - loss: 0.0091 - accuracy: 0.6928
- val_loss: 0.0088 - val_accuracy: 0.6736
Epoch 41/100
56/56 [=====] - 224s 4s/step - loss: 0.0090 - accuracy: 0.6893
- val_loss: 0.0098 - val_accuracy: 0.6517
Epoch 42/100
56/56 [=====] - 223s 4s/step - loss: 0.0091 - accuracy: 0.6803
- val_loss: 0.0090 - val_accuracy: 0.6415
Epoch 43/100
56/56 [=====] - 223s 4s/step - loss: 0.0092 - accuracy: 0.6773
- val_loss: 0.0086 - val_accuracy: 0.6867
Epoch 44/100
56/56 [=====] - 222s 4s/step - loss: 0.0090 - accuracy: 0.6793
```

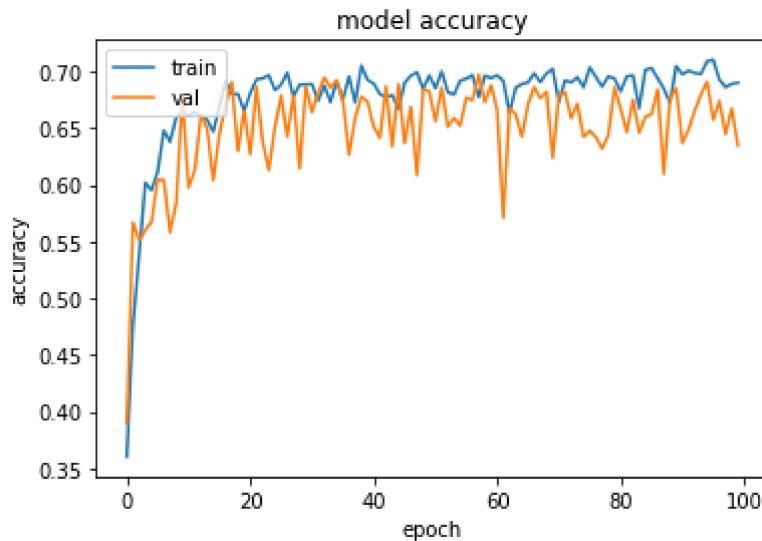
```
- val_loss: 0.0088 - val_accuracy: 0.6339
Epoch 45/100
56/56 [=====] - 224s 4s/step - loss: 0.0091 - accuracy: 0.6668
- val_loss: 0.0086 - val_accuracy: 0.6890
Epoch 46/100
56/56 [=====] - 224s 4s/step - loss: 0.0087 - accuracy: 0.6901
- val_loss: 0.0086 - val_accuracy: 0.6369
Epoch 47/100
56/56 [=====] - 222s 4s/step - loss: 0.0087 - accuracy: 0.6962
- val_loss: 0.0085 - val_accuracy: 0.6688
Epoch 48/100
56/56 [=====] - 222s 4s/step - loss: 0.0087 - accuracy: 0.6996
- val_loss: 0.0104 - val_accuracy: 0.6087
Epoch 49/100
56/56 [=====] - 222s 4s/step - loss: 0.0092 - accuracy: 0.6844
- val_loss: 0.0087 - val_accuracy: 0.6844
Epoch 50/100
56/56 [=====] - 221s 4s/step - loss: 0.0086 - accuracy: 0.6966
- val_loss: 0.0085 - val_accuracy: 0.6826
Epoch 51/100
56/56 [=====] - 222s 4s/step - loss: 0.0089 - accuracy: 0.6855
- val_loss: 0.0085 - val_accuracy: 0.6563
Epoch 52/100
56/56 [=====] - 222s 4s/step - loss: 0.0084 - accuracy: 0.7002
- val_loss: 0.0084 - val_accuracy: 0.6858
Epoch 53/100
56/56 [=====] - 228s 4s/step - loss: 0.0087 - accuracy: 0.6819
- val_loss: 0.0091 - val_accuracy: 0.6513
Epoch 54/100
56/56 [=====] - 227s 4s/step - loss: 0.0088 - accuracy: 0.6798
- val_loss: 0.0090 - val_accuracy: 0.6592
Epoch 55/100
56/56 [=====] - 227s 4s/step - loss: 0.0086 - accuracy: 0.6918
- val_loss: 0.0085 - val_accuracy: 0.6524
Epoch 56/100
56/56 [=====] - 227s 4s/step - loss: 0.0085 - accuracy: 0.6937
- val_loss: 0.0087 - val_accuracy: 0.6770
Epoch 57/100
56/56 [=====] - 228s 4s/step - loss: 0.0084 - accuracy: 0.6968
- val_loss: 0.0098 - val_accuracy: 0.6750
Epoch 58/100
56/56 [=====] - 228s 4s/step - loss: 0.0085 - accuracy: 0.6774
- val_loss: 0.0083 - val_accuracy: 0.6973
Epoch 59/100
56/56 [=====] - 226s 4s/step - loss: 0.0084 - accuracy: 0.6963
- val_loss: 0.0090 - val_accuracy: 0.6732
Epoch 60/100
56/56 [=====] - 227s 4s/step - loss: 0.0084 - accuracy: 0.6943
- val_loss: 0.0084 - val_accuracy: 0.6878
Epoch 61/100
56/56 [=====] - 228s 4s/step - loss: 0.0083 - accuracy: 0.6966
- val_loss: 0.0087 - val_accuracy: 0.6662
Epoch 62/100
56/56 [=====] - 227s 4s/step - loss: 0.0086 - accuracy: 0.6917
- val_loss: 0.0090 - val_accuracy: 0.5710
Epoch 63/100
56/56 [=====] - 226s 4s/step - loss: 0.0089 - accuracy: 0.6636
- val_loss: 0.0084 - val_accuracy: 0.6678
Epoch 64/100
56/56 [=====] - 228s 4s/step - loss: 0.0085 - accuracy: 0.6856
```

```
- val_loss: 0.0085 - val_accuracy: 0.6625
Epoch 65/100
56/56 [=====] - 227s 4s/step - loss: 0.0083 - accuracy: 0.6889
- val_loss: 0.0083 - val_accuracy: 0.6427
Epoch 66/100
56/56 [=====] - 226s 4s/step - loss: 0.0084 - accuracy: 0.6903
- val_loss: 0.0088 - val_accuracy: 0.6711
Epoch 67/100
56/56 [=====] - 228s 4s/step - loss: 0.0084 - accuracy: 0.6984
- val_loss: 0.0082 - val_accuracy: 0.6864
Epoch 68/100
56/56 [=====] - 227s 4s/step - loss: 0.0082 - accuracy: 0.6908
- val_loss: 0.0085 - val_accuracy: 0.6766
Epoch 69/100
56/56 [=====] - 226s 4s/step - loss: 0.0083 - accuracy: 0.6983
- val_loss: 0.0083 - val_accuracy: 0.6821
Epoch 70/100
56/56 [=====] - 228s 4s/step - loss: 0.0083 - accuracy: 0.7023
- val_loss: 0.0088 - val_accuracy: 0.6239
Epoch 71/100
56/56 [=====] - 227s 4s/step - loss: 0.0087 - accuracy: 0.6734
- val_loss: 0.0082 - val_accuracy: 0.6793
Epoch 72/100
56/56 [=====] - 230s 4s/step - loss: 0.0082 - accuracy: 0.6922
- val_loss: 0.0085 - val_accuracy: 0.6816
Epoch 73/100
56/56 [=====] - 230s 4s/step - loss: 0.0083 - accuracy: 0.6905
- val_loss: 0.0098 - val_accuracy: 0.6592
Epoch 74/100
56/56 [=====] - 228s 4s/step - loss: 0.0085 - accuracy: 0.6949
- val_loss: 0.0085 - val_accuracy: 0.6717
Epoch 75/100
56/56 [=====] - 228s 4s/step - loss: 0.0084 - accuracy: 0.6862
- val_loss: 0.0082 - val_accuracy: 0.6423
Epoch 76/100
56/56 [=====] - 228s 4s/step - loss: 0.0079 - accuracy: 0.7038
- val_loss: 0.0083 - val_accuracy: 0.6479
Epoch 77/100
56/56 [=====] - 229s 4s/step - loss: 0.0083 - accuracy: 0.6951
- val_loss: 0.0085 - val_accuracy: 0.6421
Epoch 78/100
56/56 [=====] - 227s 4s/step - loss: 0.0083 - accuracy: 0.6868
- val_loss: 0.0082 - val_accuracy: 0.6321
Epoch 79/100
56/56 [=====] - 227s 4s/step - loss: 0.0082 - accuracy: 0.6957
- val_loss: 0.0082 - val_accuracy: 0.6437
Epoch 80/100
56/56 [=====] - 227s 4s/step - loss: 0.0082 - accuracy: 0.6939
- val_loss: 0.0084 - val_accuracy: 0.6860
Epoch 81/100
56/56 [=====] - 227s 4s/step - loss: 0.0083 - accuracy: 0.6825
- val_loss: 0.0087 - val_accuracy: 0.6673
Epoch 82/100
56/56 [=====] - 229s 4s/step - loss: 0.0083 - accuracy: 0.6958
- val_loss: 0.0080 - val_accuracy: 0.6468
Epoch 83/100
56/56 [=====] - 228s 4s/step - loss: 0.0080 - accuracy: 0.6966
- val_loss: 0.0086 - val_accuracy: 0.6751
Epoch 84/100
56/56 [=====] - 233s 4s/step - loss: 0.0082 - accuracy: 0.6677
```

```
- val_loss: 0.0081 - val_accuracy: 0.6460
Epoch 85/100
56/56 [=====] - 230s 4s/step - loss: 0.0078 - accuracy: 0.7014
- val_loss: 0.0089 - val_accuracy: 0.6601
Epoch 86/100
56/56 [=====] - 230s 4s/step - loss: 0.0079 - accuracy: 0.7031
- val_loss: 0.0090 - val_accuracy: 0.6624
Epoch 87/100
56/56 [=====] - 230s 4s/step - loss: 0.0081 - accuracy: 0.6941
- val_loss: 0.0079 - val_accuracy: 0.6843
Epoch 88/100
56/56 [=====] - 228s 4s/step - loss: 0.0079 - accuracy: 0.6849
- val_loss: 0.0087 - val_accuracy: 0.6097
Epoch 89/100
56/56 [=====] - 228s 4s/step - loss: 0.0082 - accuracy: 0.6725
- val_loss: 0.0081 - val_accuracy: 0.6764
Epoch 90/100
56/56 [=====] - 227s 4s/step - loss: 0.0080 - accuracy: 0.7048
- val_loss: 0.0081 - val_accuracy: 0.6853
Epoch 91/100
56/56 [=====] - 228s 4s/step - loss: 0.0081 - accuracy: 0.6976
- val_loss: 0.0086 - val_accuracy: 0.6371
Epoch 92/100
56/56 [=====] - 226s 4s/step - loss: 0.0079 - accuracy: 0.7007
- val_loss: 0.0083 - val_accuracy: 0.6489
Epoch 93/100
56/56 [=====] - 229s 4s/step - loss: 0.0077 - accuracy: 0.6988
- val_loss: 0.0080 - val_accuracy: 0.6656
Epoch 94/100
56/56 [=====] - 228s 4s/step - loss: 0.0078 - accuracy: 0.6978
- val_loss: 0.0083 - val_accuracy: 0.6796
Epoch 95/100
56/56 [=====] - 228s 4s/step - loss: 0.0079 - accuracy: 0.7089
- val_loss: 0.0079 - val_accuracy: 0.6910
Epoch 96/100
56/56 [=====] - 228s 4s/step - loss: 0.0077 - accuracy: 0.7105
- val_loss: 0.0081 - val_accuracy: 0.6576
Epoch 97/100
56/56 [=====] - 227s 4s/step - loss: 0.0080 - accuracy: 0.6929
- val_loss: 0.0082 - val_accuracy: 0.6742
Epoch 98/100
56/56 [=====] - 227s 4s/step - loss: 0.0080 - accuracy: 0.6862
- val_loss: 0.0087 - val_accuracy: 0.6449
Epoch 99/100
56/56 [=====] - 228s 4s/step - loss: 0.0079 - accuracy: 0.6891
- val_loss: 0.0080 - val_accuracy: 0.6673
Epoch 100/100
56/56 [=====] - 230s 4s/step - loss: 0.0080 - accuracy: 0.6901
- val_loss: 0.0082 - val_accuracy: 0.6349
```

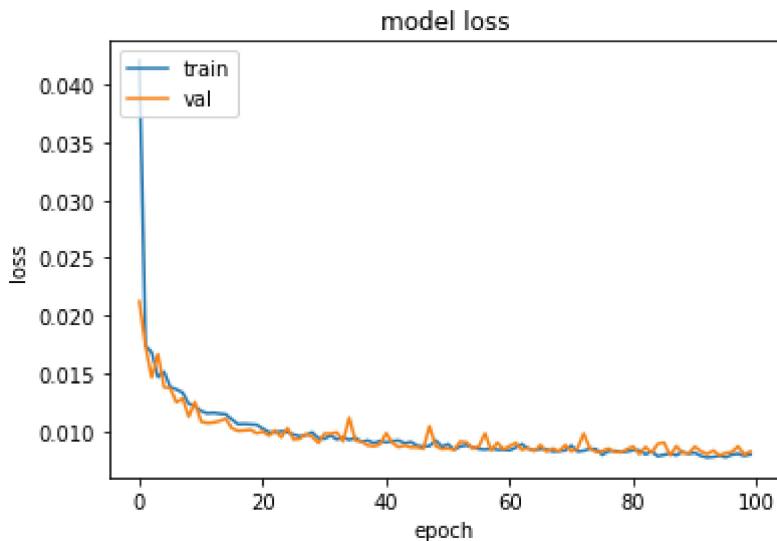
In [14]:

```
plt.plot(history.history['accuracy'])
plt.plot(history.history['val_accuracy'])
plt.title('model accuracy')
plt.ylabel('accuracy')
plt.xlabel('epoch')
plt.legend(['train', 'val'], loc='upper left')
plt.show()
```



In [15]:

```
plt.plot(history.history['loss'])
plt.plot(history.history['val_loss'])
plt.title('model loss')
plt.ylabel('loss')
plt.xlabel('epoch')
plt.legend(['train', 'val'], loc='upper left')
plt.show()
```



In [16]:

```
autoencoder.save("./AI_Denoising_WeightsAfter100Epochs3layers512size")
!zip -r /content/AI_Denoising_WeightsAfter100Epochs3layers512size.zip /content/AI_Deno
```

2021-11-29 05:34:04.248375: W tensorflow/python/util/util.cc:368] Sets are not currently considered sequences, but this may change in the future, so consider avoiding using the m.

INFO:tensorflow:Assets written to: ./AI\_Denoising\_WeightsAfter100Epochs3layers512size/assets

/bin/bash: zip: command not found

In [17]:

```
autoencoder.save("./DenoisingModel512Size3Layers")
```

INFO:tensorflow:Assets written to: ./DenoisingModel512Size3Layers/assets

```
In [19]: test_rain_images = []
for i in range(58):
    file = "Dataset/Test/data/" + str(i) + "_rain.png"
    test_rain_images.append(cv2.imread(file))
```

```
In [20]: test_clean_images = []
for i in range(58):
    file = "Dataset/Test/gt/" + str(i) + "_clean.png"
    test_clean_images.append(cv2.imread(file))
```

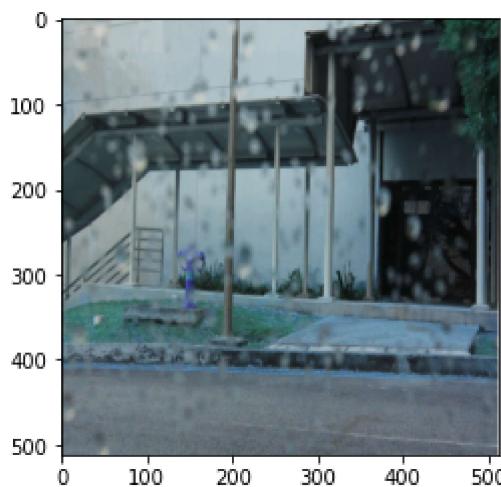
```
In [21]: test_rain_final_images = []
for i in test_rain_images:
    x = cv2.resize(i,(512,512))
    x = x/255
    test_rain_final_images.append(x)

test_clean_final_images = []
for i in test_clean_images:
    x = cv2.resize(i,(512,512))
    x = x/255
    test_clean_final_images.append(x)
```

```
In [22]: test_rain_final_images = np.asarray(test_rain_final_images)
test_rain_final_images.shape
test_derained_final_images = autoencoder.predict(test_rain_final_images)
```

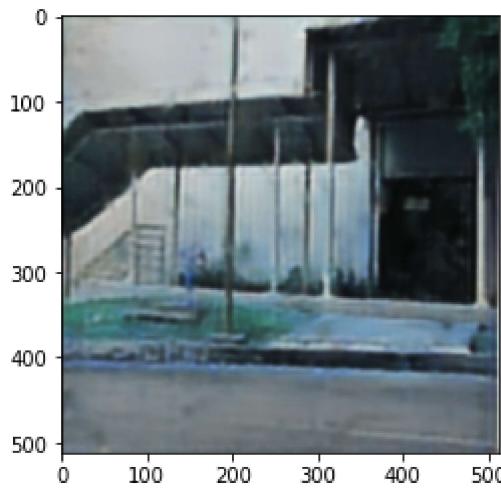
```
In [32]: plt.imshow(test_rain_final_images[21])
```

Out[32]: <matplotlib.image.AxesImage at 0x7fe6942ae130>



```
In [31]: plt.imshow(test_derained_final_images[21])
```

Out[31]: <matplotlib.image.AxesImage at 0x7fe668673dc0>



```
In [25]: mse = ((test_clean_final_images - test_derained_final_images) ** 2).mean(axis=None)  
mse
```

```
Out[25]: 0.004305285439150233
```

```
In [26]: psnr = 20*np.log10(1/(mse** (1/2.0)))  
psnr
```

```
Out[26]: 23.659980496601833
```

```
In [ ]:
```