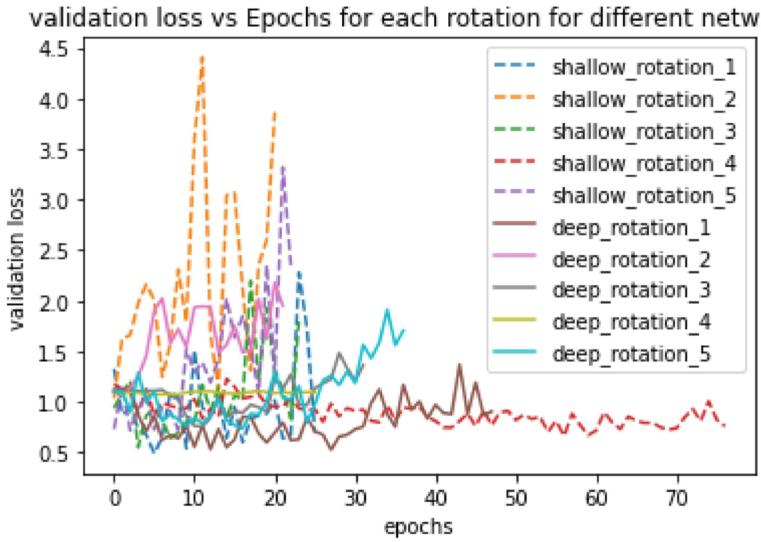
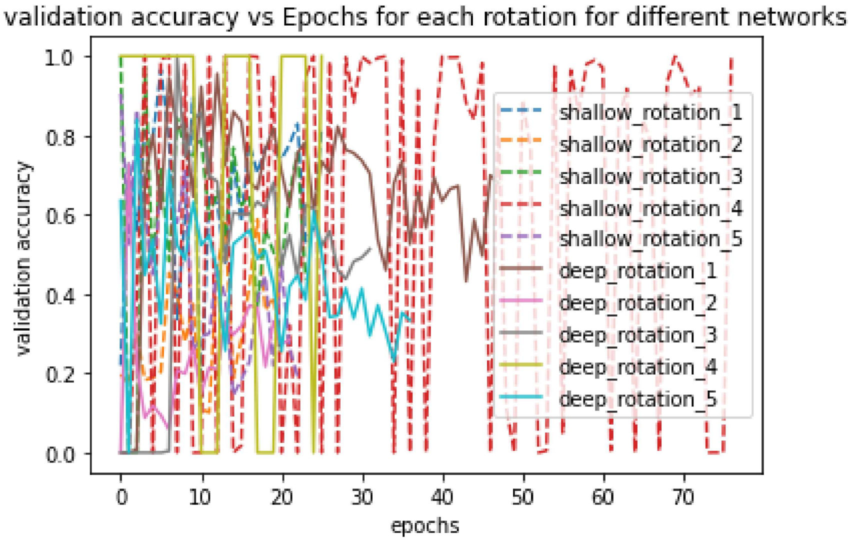
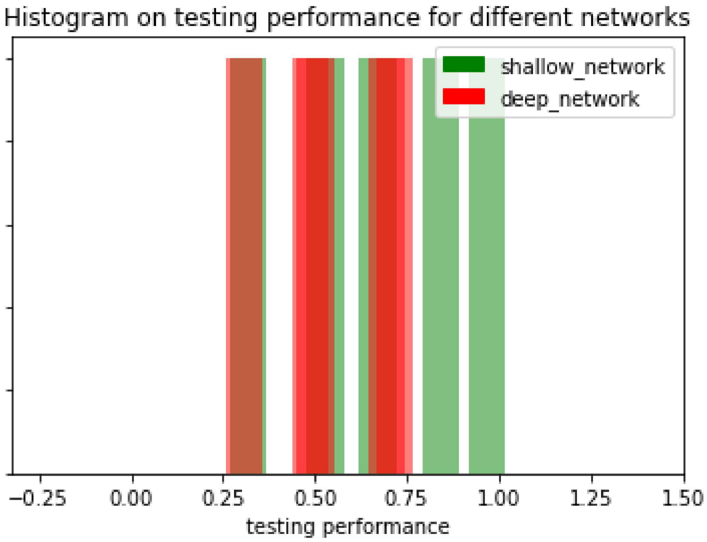
In [71]:

```
import matplotlib.pyplot as plt
import tensorflow as tf
import pandas as pd
import numpy as np
from tensorflow import keras
import os
import fnmatch
import time
import pickle
from tensorflow.keras.layers import InputLayer, Dense
from tensorflow.keras.models import Sequential
import matplotlib.patches as mpatches

def read_all_rotations(dirname, filebase):
    '''Read results from dirname from files matching filebase'''

    # The set of files in the directory
    files = fnmatch.filter(os.listdir(dirname), filebase)
    files.sort()
    results = []


    # Loop over matching files
    for f in files:
        fp = open("%s/%s"%(dirname,f), "rb")
        r = pickle.load(fp)
        fp.close()
        results.append(r)
    return results

#shallow network
filebase = "image_Csize_3_3_3_3_3_1_1_Cfilters_32_64_64_128_256_512_512_Pool_2_2_2_2_2
new_res = read_all_rotations("results1", filebase)

#deep network
#filebase1 = "image_Csize_3_3_Cfilters_16_16_Pool_2_2_hidden_35_30_25_20_drop_0.500_L2
filebase1 = "image_Csize_3_3_1_Cfilters_16_16_16_Pool_2_2_1_hidden_40_30_20_drop_0.500
new_res1 = read_all_rotations("results2", filebase1)

print(new_res[0]['predict_testing_eval'])
#figure1
for i in range(len(new_res)):
    plt.plot(new_res[i]['history']['val_loss'],linestyle='dashed',label = 'shallow_rot
for i in range(len(new_res1)):
    plt.plot(new_res1[i]['history']['val_loss'],label = 'deep_rotation_'+str(i+1))

plt.ylabel('validation loss')
plt.xlabel('epochs')
plt.title('validation loss vs Epochs for each rotation for different networks')
plt.legend()
plt.savefig("validation_loss.png")
plt.show()
plt.close()

#figure2
for i in range(len(new_res)):
    plt.plot(new_res[i]['history']['val_categorical_accuracy'],linestyle='dashed',labe
for i in range(len(new_res)):
    plt.plot(new_res1[i]['history']['val_categorical_accuracy'],label = 'deep_rotatior

plt.ylabel('validation accuracy')
plt.xlabel('epochs')
plt.title('validation accuracy vs Epochs for each rotation for different networks')
plt.legend()
plt.savefig("validation_accuracy.png")
plt.show()
plt.close()

#figure3
for i in range(len(new_res)):
    plt.hist(new_res[i]['predict_testing_eval'][1], alpha=0.5, color='green',label='sh
for i in range(len(new_res1)):
    plt.hist(new_res1[i]['predict_testing_eval'][1], alpha=0.5, color='red',label='dee
plt.xlabel('testing performance')
plt.title('Histogram on testing performance for different networks')
a = mpatches.Patch(color='green', label='shallow_network')
b = mpatches.Patch(color='red', label='deep_network')
plt.legend(handles=[a,b])
plt.savefig("histogram_accuracy.png")
plt.show()
plt.close()
```

```
[2.040663719177246, 0.4793689250946045]
```







In [ ]:

```
1. How many parameters were needed by your shallow and deep networks?
shallow networks = 1,009,637
deep networks = 580,993

2. What can you conclude from the validation accuracy learning curves for each of the
deep networks? How confident are you that you have created models that you can trust?

From the graph I can deduce that the validation accuracy for shallow networks is bette
dense networks. But the both the networks need lots of hyper-parameter tuning since sc
up pretty badly.

3. Did your shallow or deep network perform better with respect to the test set?
The shallow network performed well with respect to the test set.
```