Link to Videos: https://drive.google.com/drive/folders/1GyXrXFuu8F4rabLdEyFHLxs07XF2KzWu?usp=sharing

In [21]:
```
import pickle
import pandas as pd
import matplotlib.pyplot as plt
from copy import deepcopy
import numpy as np
import cv2
from IPython.display import Image
```

Question 1

Associated Videos: Q1_yolo_normal.mp4, Q1_yolo_Stopsign1.mp4, Q1_yolo_Stopsign2.mp4 in the provided google folder

A. YOLO Model

In [22]:
```
# c2 Question 1 Stop sign Yolo
# load the data from pkl file
with open('log\\exp\\Q1_stopsign_og_Yolo\\eval_results\\results.pkl', 'rb') as f:
    data = pickle.load(f)

# convert the data to pandas dataframe
df = pd.DataFrame(data)
print(df)
```

```
   mean_iou  mAP_evaluate  stdAP_evaluate
0  0.747807      0.485362        0.375350
1  0.837157      0.702872        0.317317
2  0.878591      0.788699        0.311963
3  0.872948      0.789422        0.290203
```

In [23]:
```
# c2 Question 1 Stop sign 1 Yolo
# load the data from pkl file
with open('log\\exp\\Q1_stopsign_1_Yolo\\eval_results\\results.pkl', 'rb') as f:
    data = pickle.load(f)

# convert the data to pandas dataframe
df = pd.DataFrame(data)
print(df)
```

```
   mean_iou  mAP_evaluate  stdAP_evaluate
0  0.580653      0.376966        0.278740
1  0.610379      0.529526        0.199966
2  0.850288      0.729941        0.333669
3  0.842384      0.717358        0.312095
```

In [24]:
```
# c2 Question 1 Stop sign 2 Yolo
# load the data from pkl file
with open('log\\exp\\Q1_stopsign_2_Yolo\\eval_results\\results.pkl', 'rb') as f:
    data = pickle.load(f)

# convert the data to pandas dataframe
df = pd.DataFrame(data)
print(df)
```

```
   mean_iou  mAP_evaluate  stdAP_evaluate
0  0.531654      0.368143        0.275532
1  0.626392      0.549553        0.178625
```

```
2   0.843756      0.718918        0.329150
3   0.770554      0.664913        0.272914
```

B. FRCNN

In [25]:
```
# c2 Question 1 Stop sign FRCNN
# load the data from pkl file
with open('log\\exp\\Q1_stopsign_og_FRCNN\\eval_results\\results.pkl', 'rb') as f:
    data = pickle.load(f)

# convert the data to pandas dataframe
df = pd.DataFrame(data)
print(df)
```

```
   mean_iou  mAP_evaluate  stdAP_evaluate
0  0.364685      0.207429        0.156719
1  0.429130      0.403049        0.112454
2  0.608553      0.557508        0.168154
3  0.581772      0.522839        0.168156
```

In [26]:
```
# c2 Question 1 Stop sign 1 FRCNN
# load the data from pkl file
with open('log\\exp\\Q1_stopsign_1_FRCNN\\eval_results\\results.pkl', 'rb') as f:
    data = pickle.load(f)

# convert the data to pandas dataframe
df = pd.DataFrame(data)
print(df)
```

```
   mean_iou  mAP_evaluate  stdAP_evaluate
0  0.331755      0.174533        0.163422
1  0.396887      0.355946        0.122129
2  0.442069      0.395739        0.141531
3  0.401119      0.353856        0.141964
```

In [27]:
```
# c2 Question 1 Stop sign FRCNN
# load the data from pkl file
with open('log\\exp\\Q1_stopsign_2_FRCNN\\eval_results\\results.pkl', 'rb') as f:
    data = pickle.load(f)

# convert the data to pandas dataframe
df = pd.DataFrame(data)
print(df)
```

```
   mean_iou  mAP_evaluate  stdAP_evaluate
0  0.348696      0.184400        0.144876
1  0.393145      0.328464        0.117536
2  0.715579      0.573095        0.248651
3  0.490226      0.397055        0.161647
```

Yolo performs significantly better than faster rcnn when we look at the mean. The results are more robust charecterised by a smaller standard deviation. However, computation time is slightly longer with Yolo.

Question 2

Associated videos starting with Q2 in the provided Google folder

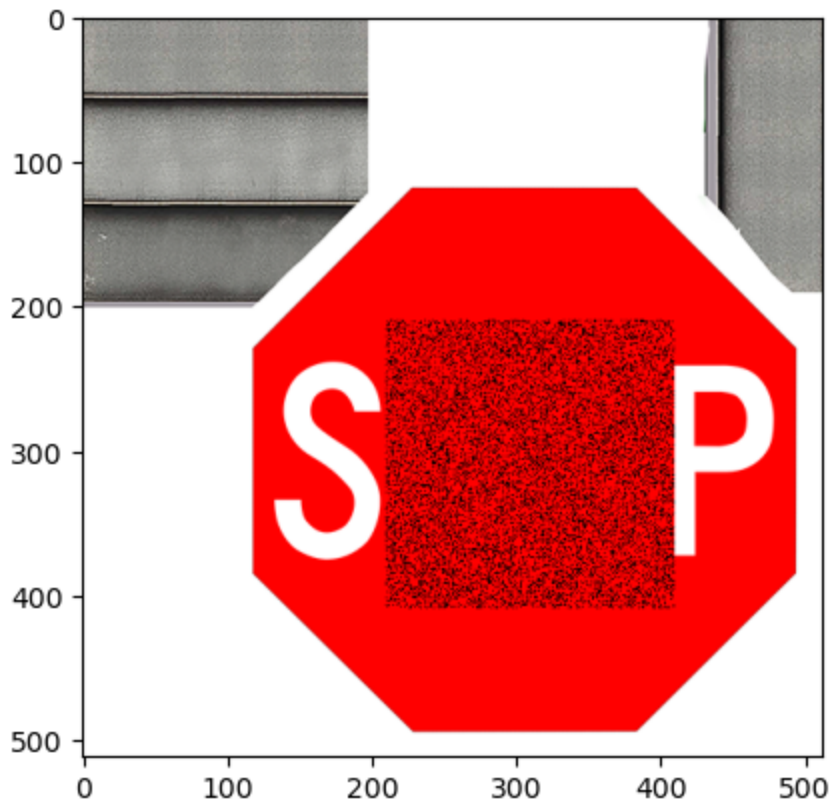
Patch Version 1 (Best Performing)

In [28]:
```
root = 'safebench/scenario/scenario_data/template_od/'
source_img = cv2.imread(root + 'stopsign.jpg')
```

```
sign_center = (310,310)
patch_size = 200

# find corners of the patch
top_left = (sign_center[0] - patch_size//2, sign_center[1] - patch_size//2)
bottom_right = (sign_center[0] + patch_size//2, sign_center[1] + patch_size//2)


# Generate random pixels between blue and white
random_pixels = np.ones((200, 200, 3), dtype=np.uint8) * [0, 0, 0]

# Randomly convert some pixels to red
for i in range(200):
    for j in range(200):
        # Generate a random number between 0 and 1
        random_number = np.random.rand()

        # If the random number is less than 0.1, convert the pixel to red
        if random_number < 0.75:
            random_pixels[i, j] = [0, 0, 255]

# Convert the random pixels type to uint8
random_pixels = random_pixels.astype(np.uint8)
# plt.imshow(cv2.cvtColor(random_pixels, cv2.COLOR_BGR2RGB))

# Put the random pixels on the image
stop_img = deepcopy(source_img)
stop_img[top_left[1]:bottom_right[1], top_left[0]:bottom_right[0]] = random_pixels
print(stop_img.shape)
# Display the image
plt.imshow(cv2.cvtColor(stop_img, cv2.COLOR_BGR2RGB))
```

Out[28]:

```
(512, 512, 3)
<matplotlib.image.AxesImage at 0x2161ff78fd0>
```



In [29]:

```
# c2 Question 2 Stop sign patch 7 Yolo
# load the data from pkl file
with open('log\\exp\\Q2_patch7_Yolo\\eval_results\\results.pkl', 'rb') as f:
```

```
    data = pickle.load(f)

# convert the data to pandas dataframe
df = pd.DataFrame(data)
print(df)
```

```
   mean_iou  mAP_evaluate  stdAP_evaluate
0  0.419450      0.255224        0.203216
1  0.473348      0.432959        0.125391
2  0.851177      0.739633        0.294254
3  0.639012      0.553370        0.186233
```

In [30]:
```
# c2 Question 2 Stop sign patch 7 FRCNN
# load the data from pkl file
with open('log\\exp\\Q2_patch7_FRCNN\\eval_results\\results.pkl', 'rb') as f:
    data = pickle.load(f)

# convert the data to pandas dataframe
df = pd.DataFrame(data)
print(df)
```

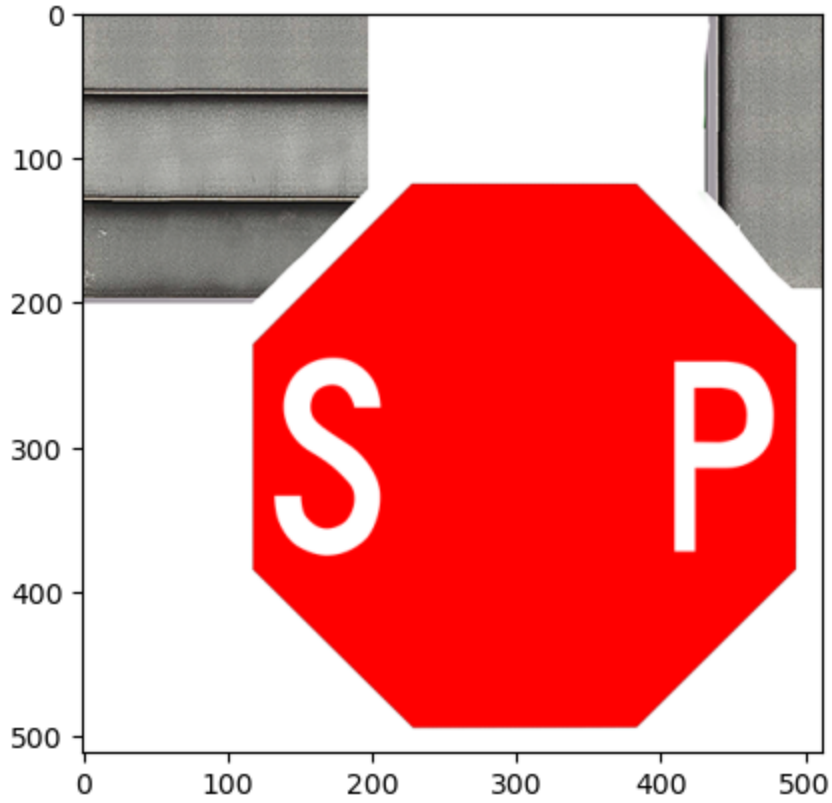
```
   mean_iou  mAP_evaluate  stdAP_evaluate
0  0.346788      0.190702        0.161529
1  0.407909      0.379969        0.095941
2  0.515926      0.475723        0.138329
3  0.477053      0.413670        0.147604
```

This is the patch which performs best. We also attach another patch which performs well

Patch Version 2 (Random)

In [31]:
```
root = 'safebench/scenario/scenario_data/template_od/'
source_img = cv2.imread(root + 'stopsign.jpg')

sign_center = (310,310)
patch_size = 200

# find corners of the patch
top_left = (sign_center[0] - patch_size//2, sign_center[1] - patch_size//2)
bottom_right = (sign_center[0] + patch_size//2, sign_center[1] + patch_size//2)


patch = np.ones((200, 200, 3), dtype=np.uint8) * [0, 0, 255]
stop_img = deepcopy(source_img)
stop_img[top_left[1]:bottom_right[1], top_left[0]:bottom_right[0]] = patch

# display the image
plt.imshow(cv2.cvtColor(stop_img, cv2.COLOR_BGR2RGB))
```

Out[31]:
```
<matplotlib.image.AxesImage at 0x2161ffd8b20>
```



In [32]:
```
# c2 Question 2 Stop sign patch 4 Yolo
# load the data from pkl file
with open('log\\exp\\Q2_patch4_Yolo\\eval_results\\results.pkl', 'rb') as f:
    data = pickle.load(f)

# convert the data to pandas dataframe
df = pd.DataFrame(data)
print(df)
```

```
   mean_iou  mAP_evaluate  stdAP_evaluate
0  0.450400      0.271723        0.223114
1  0.474684      0.438677        0.123328
2  0.844314      0.725846        0.303676
3  0.789530      0.683827        0.269486
```

In [33]:
```
# c2 Question 2 Stop sign patch 4 FRCNN
# load the data from pkl file
with open('log\\exp\\Q2_patch4_FRCNN\\eval_results\\results.pkl', 'rb') as f:
    data = pickle.load(f)

# convert the data to pandas dataframe
df = pd.DataFrame(data)
print(df)
```

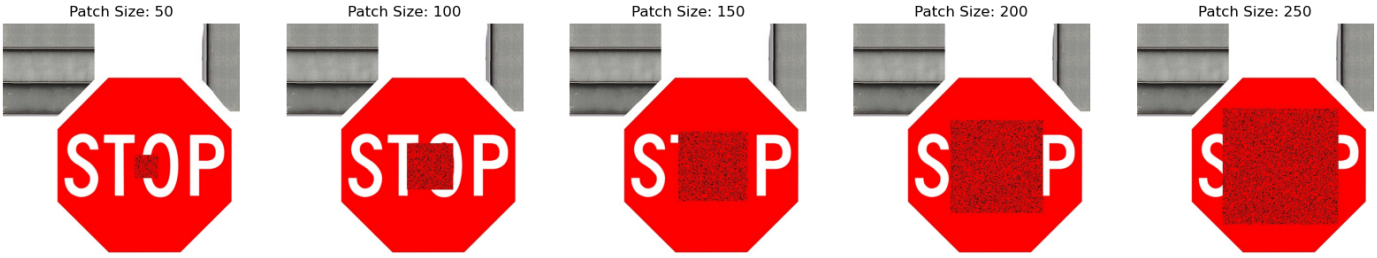
```
   mean_iou  mAP_evaluate  stdAP_evaluate
0  0.348039      0.193215        0.159601
1  0.415259      0.382362        0.100020
2  0.548646      0.498912        0.158705
3  0.529803      0.446598        0.166322
```

Challenge 3

Scaling the patch from size 100 to 300 in increments of 50

Associated videos with Q3_s{patch_size}.mp4 in the provided Google Folder

In [34]:
```
root = 'safebench/scenario/scenario_data/template_od/'

img_list = []

for i in [50,100,150,200,250]:
    img_list.append(cv2.imread(root + 'stopsign_q3_scale' + str(i) + '.jpg'))

# Display the images in a 1x5 grid
fig, ax = plt.subplots(1, 5, figsize=(20, 20))
for i in range(5):
    ax[i].imshow(cv2.cvtColor(img_list[i], cv2.COLOR_BGR2RGB))
    ax[i].title.set_text('Patch Size: ' + str((i+1)*50))
    ax[i].axis('off')
plt.show()
```



A. Running these patches with YOLO

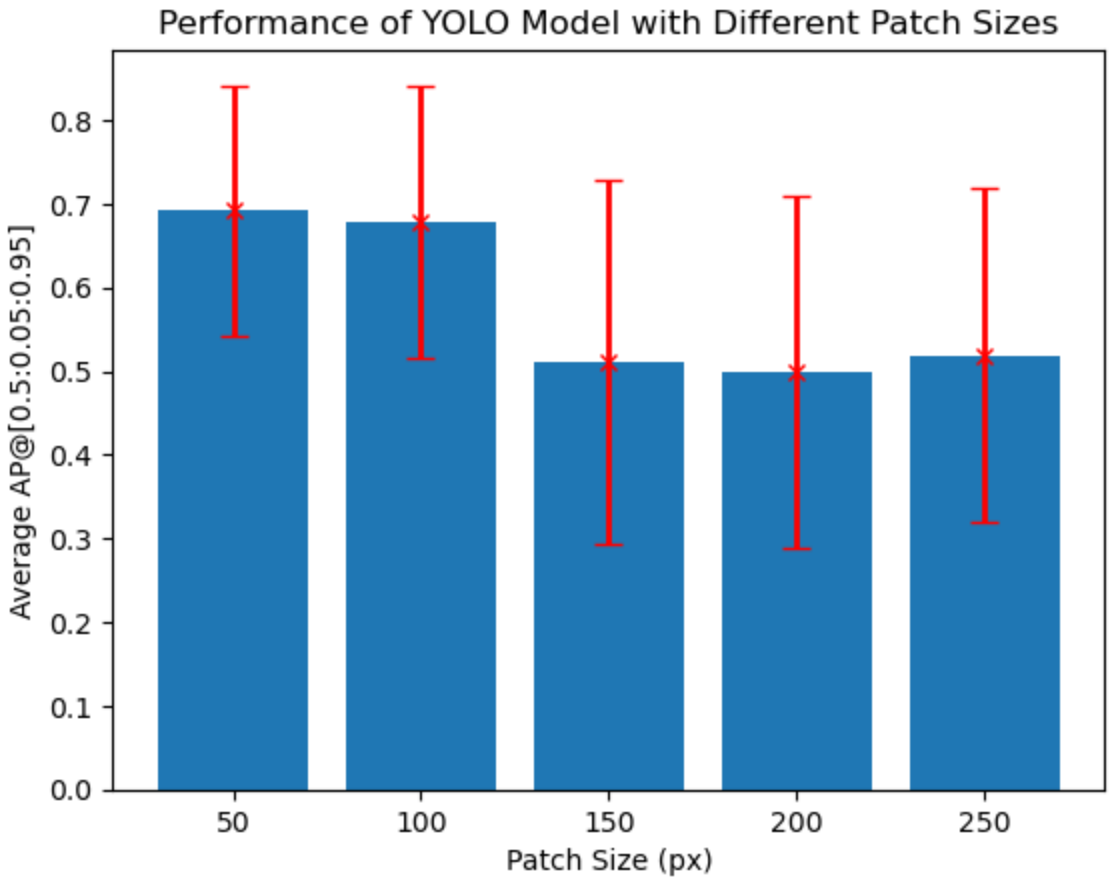
In [58]:
```
root = 'log'

result_list = []
history_list_mean = []
history_list_std = []

for i in [50,100,150,200,250]:
    with open(root + f'\\scale_{i}_yolo\\scale_{i}_yolo_seed_0\\eval_results\\results.pk
        data = pickle.load(f)
    df = pd.DataFrame(data)
    print(f"Patch Size {i} used with YOLO Model")
    print(df)
    print('\n')
    result_list.append(df)
    history_list_mean.append(df["mAP_evaluate"].mean())
    history_list_std.append(df["mAP_evaluate"].std())

# Bar plot for mAP
plt.bar(["50","100","150","200","250"], history_list_mean)
plt.errorbar(["50","100","150","200","250"], history_list_mean, yerr=history_list_std, f
plt.xlabel('Patch Size (px)')
plt.ylabel('Average AP@[0.5:0.05:0.95]')
plt.title('Performance of YOLO Model with Different Patch Sizes')
```

```
Patch Size 50 used with YOLO Model
   mean_iou  mAP_evaluate  stdAP_evaluate
0  0.747622      0.481956        0.380052
1  0.829449      0.687668        0.303688
2  0.881742      0.798086        0.296445
3  0.878255      0.799593        0.282435


Patch Size 100 used with YOLO Model
   mean_iou  mAP_evaluate  stdAP_evaluate
0  0.729828      0.456817        0.365838
1  0.809605      0.658455        0.299207
2  0.886371      0.805699        0.296282
3  0.873319      0.791594        0.281312
```

```
Patch Size 150 used with YOLO Model
   mean_iou  mAP_evaluate  stdAP_evaluate
0  0.413042      0.252257        0.201091
1  0.480783      0.435492        0.128661
2  0.861993      0.759459        0.303148
3  0.683428      0.595836        0.193856


Patch Size 200 used with YOLO Model
   mean_iou  mAP_evaluate  stdAP_evaluate
0  0.411451      0.244033        0.195309
1  0.483073      0.445665        0.117756
2  0.846498      0.746698        0.290589
3  0.645604      0.556563        0.191514


Patch Size 250 used with YOLO Model
   mean_iou  mAP_evaluate  stdAP_evaluate
0  0.452073      0.273114        0.220452
1  0.495837      0.456333        0.130949
2  0.848935      0.731698        0.308760
3  0.712126      0.614743        0.238345
```

Out[58]:

```
Text(0.5, 1.0, 'Performance of YOLO Model with Different Patch Sizes')
```



B. Running these patches with Faster RCNN

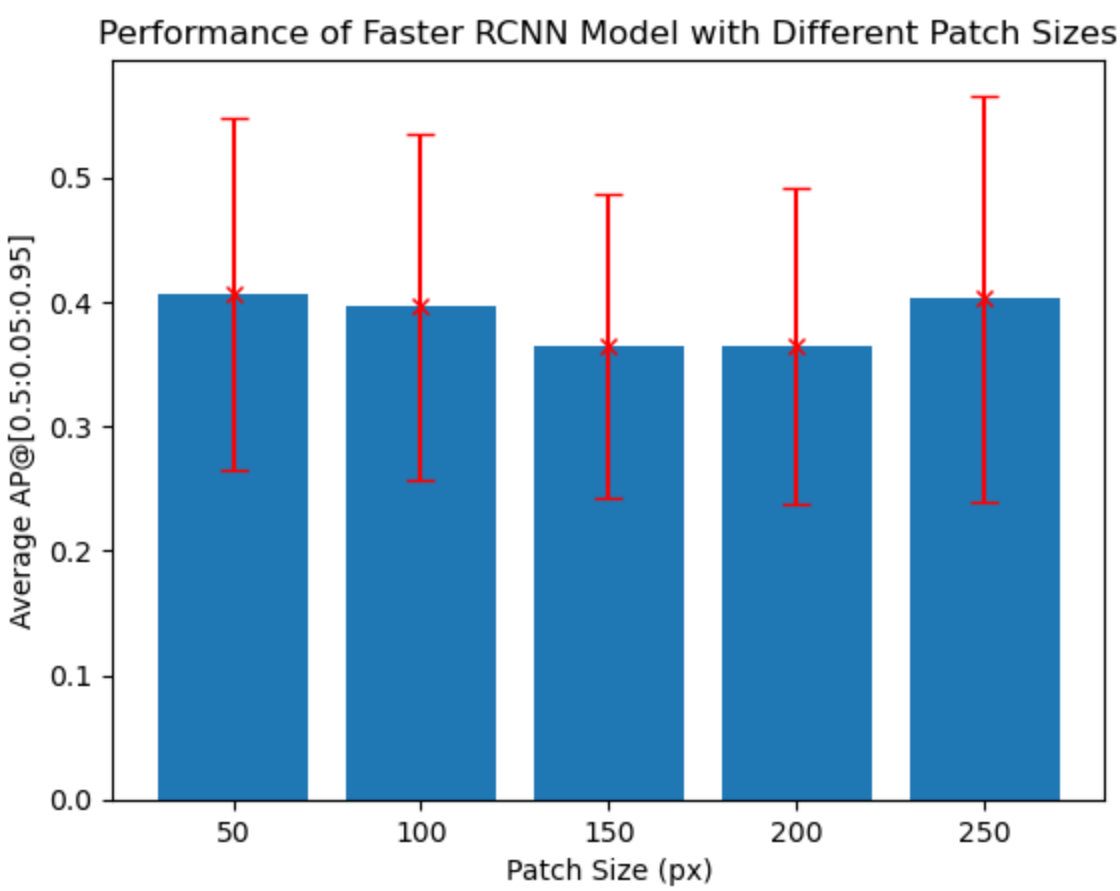
In [57]:

```
root = 'log'

result_list = []
history_list_mean = []
history_list_std = []
```

```
for i in [50,100,150,200,250]:
    with open(root + f'\\scale_{i}_frcnn\\scale_{i}_frcnn_seed_0\\eval_results\\results.
        data = pickle.load(f)
    df = pd.DataFrame(data)
    print(f"Patch Size {i} used with Faster RCNN Model")
    print(df)
    print("\n")
    result_list.append(df)
    history_list_mean.append(df["mAP_evaluate"].mean())
    history_list_std.append(df["mAP_evaluate"].std())

# Bar plot for mAP
plt.bar(["50","100","150","200","250"], history_list_mean)
plt.errorbar(["50","100","150","200","250"], history_list_mean, yerr=history_list_std, f
plt.xlabel('Patch Size (px)')
plt.ylabel('Average AP@[0.5:0.05:0.95]')
plt.title('Performance of Faster RCNN Model with Different Patch Sizes')
```

```
Patch Size 50 used with Faster RCNN Model
   mean_iou  mAP_evaluate  stdAP_evaluate
0  0.364271      0.210521        0.158436
1  0.419796      0.393397        0.112927
2  0.547508      0.511194        0.139141
3  0.571000      0.510255        0.167567


Patch Size 100 used with Faster RCNN Model
   mean_iou  mAP_evaluate  stdAP_evaluate
0  0.363508      0.202804        0.157551
1  0.418694      0.389903        0.111595
2  0.549471      0.514376        0.136752
3  0.523351      0.477394        0.147561


Patch Size 150 used with Faster RCNN Model
   mean_iou  mAP_evaluate  stdAP_evaluate
0  0.349269      0.189849        0.158756
1  0.405588      0.375927        0.100484
2  0.506626      0.469583        0.138026
3  0.471870      0.422139        0.145732


Patch Size 200 used with Faster RCNN Model
   mean_iou  mAP_evaluate  stdAP_evaluate
0  0.339277      0.183007        0.153502
1  0.405378      0.376512        0.101249
2  0.515274      0.473935        0.136851
3  0.478861      0.422206        0.148599


Patch Size 250 used with Faster RCNN Model
   mean_iou  mAP_evaluate  stdAP_evaluate
0  0.347309      0.188093        0.156815
1  0.406394      0.374685        0.094463
2  0.645003      0.566360        0.220808
3  0.565556      0.482019        0.187962
```

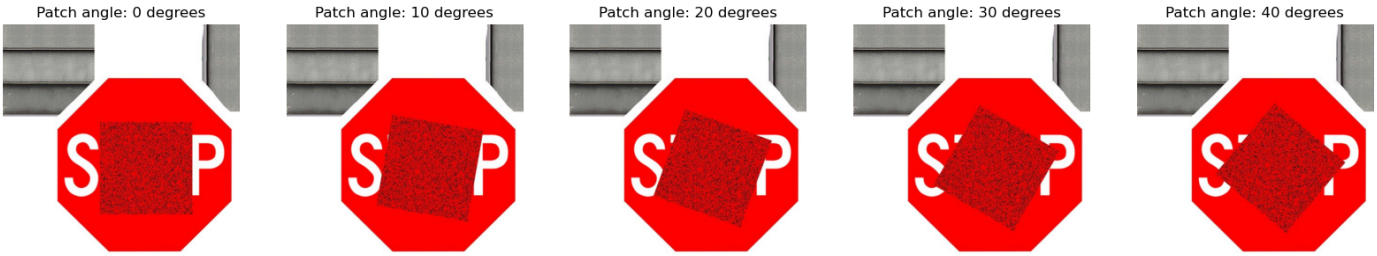
Out[57]:
```
Text(0.5, 1.0, 'Performance of Faster RCNN Model with Different Patch Sizes')
```



Rotating the patch from 0 to 40 degrees in increments of 10

Associated videos with Q3_r{patch_angle}.mp4 in the provided Google Folder

In [62]:
```
root = 'safebench/scenario/scenario_data/template_od/'

img_list = []

for i in [0,10,20,30,40]:
    img_list.append(cv2.imread(root + 'stopsign_q3_rotate' + str(i) + '.jpg'))

# Display the images in a 1x5 grid
fig, ax = plt.subplots(1, 5, figsize=(20, 20))
for i in range(5):
    ax[i].imshow(cv2.cvtColor(img_list[i], cv2.COLOR_BGR2RGB))
    ax[i].title.set_text('Patch angle: ' + str(i*10) + ' degrees')
    ax[i].axis('off')
plt.show()
```



A. Running these patches with YOLO

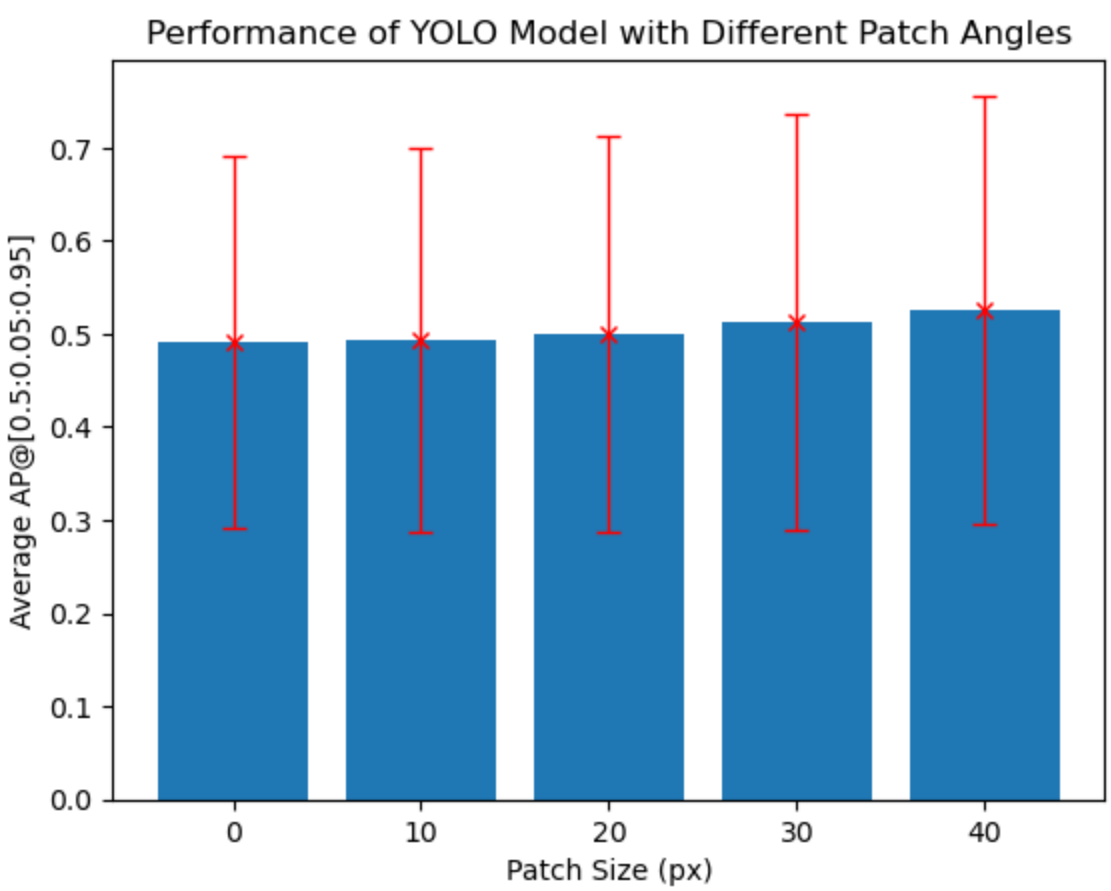
In [59]:
```
root = 'log\\exp'

result_list = []
```

```
history_list_mean = []
history_list_std = []

for i in [0,10,20,30,40]:
    with open(root + f'\\Q3_r{i}_yolo\\eval_results\\results.pkl', 'rb') as f:
        data = pickle.load(f)
    df = pd.DataFrame(data)
    print(f"Patch angle {i} used with YOLO Model")
    print(df)
    print('\n')
    result_list.append(df)
    history_list_mean.append(df["mAP_evaluate"].mean())
    history_list_std.append(df["mAP_evaluate"].std())

# Bar plot for mAP
plt.bar(["0","10","20","30","40"], history_list_mean)
plt.errorbar(["0","10","20","30","40"], history_list_mean, yerr=history_list_std, fmt='x
plt.xlabel('Patch Size (px)')
plt.ylabel('Average AP@[0.5:0.05:0.95]')
plt.title('Performance of YOLO Model with Different Patch Angles')
```

```
Patch angle 0 used with YOLO Model
   mean_iou  mAP_evaluate  stdAP_evaluate
0  0.426848      0.254600        0.205329
1  0.484635      0.448520        0.125638
2  0.850229      0.739180        0.307089
3  0.602023      0.519987        0.169259


Patch angle 10 used with YOLO Model
   mean_iou  mAP_evaluate  stdAP_evaluate
0  0.403281      0.240207        0.192148
1  0.484654      0.445150        0.129489
2  0.849254      0.734186        0.304680
3  0.632953      0.550028        0.191050


Patch angle 20 used with YOLO Model
   mean_iou  mAP_evaluate  stdAP_evaluate
0  0.403251      0.242274        0.189279
1  0.483850      0.440194        0.136612
2  0.855081      0.749628        0.300218
3  0.649583      0.567040        0.209382


Patch angle 30 used with YOLO Model
   mean_iou  mAP_evaluate  stdAP_evaluate
0  0.411392      0.240310        0.191631
1  0.483404      0.436591        0.135664
2  0.863826      0.758856        0.308816
3  0.717620      0.614915        0.226848


Patch angle 40 used with YOLO Model
   mean_iou  mAP_evaluate  stdAP_evaluate
0  0.420308      0.248687        0.195461
1  0.487302      0.442726        0.126148
2  0.870588      0.775575        0.300931
3  0.724933      0.637542        0.235527
```

Out[59]:

```
Text(0.5, 1.0, 'Performance of YOLO Model with Different Patch Angles')
```



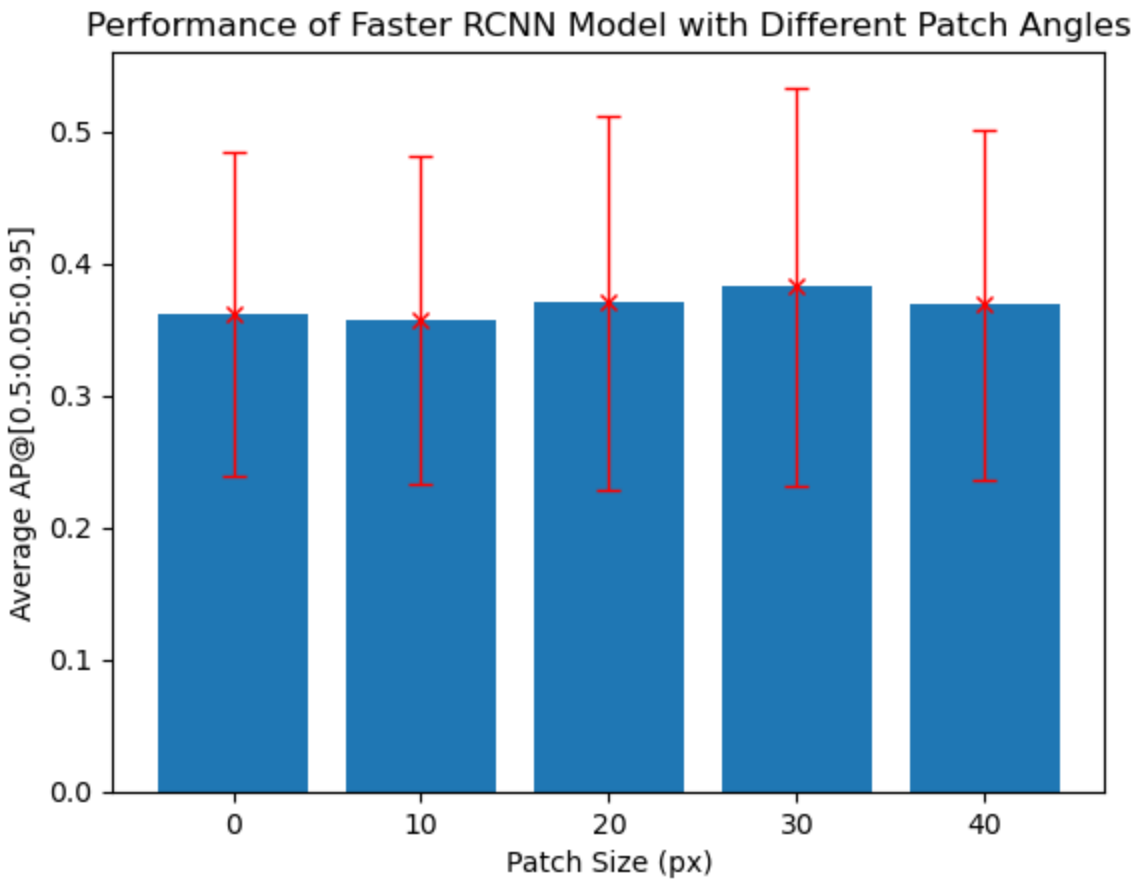
B. Running these patches with Faster RCNN

In [60]:
```
root = 'log\\exp'

result_list = []
history_list_mean = []
history_list_std = []

for i in [0,10,20,30,40]:
    with open(root + f'\\Q3_r{i}_frcnn\\eval_results\\results.pkl', 'rb') as f:
        data = pickle.load(f)
    df = pd.DataFrame(data)
    print(f"Patch angle {i} used with Faster RCNN Model")
    print(df)
    print('\n')
    result_list.append(df)
    history_list_mean.append(df["mAP_evaluate"].mean())
    history_list_std.append(df["mAP_evaluate"].std())

# Bar plot for mAP
plt.bar(["0","10","20","30","40"], history_list_mean)
plt.errorbar(["0","10","20","30","40"], history_list_mean, yerr=history_list_std, fmt='x
plt.xlabel('Patch Size (px)')
plt.ylabel('Average AP@[0.5:0.05:0.95]')
plt.title('Performance of Faster RCNN Model with Different Patch Angles')
```

```
Patch angle 0 used with Faster RCNN Model
   mean_iou  mAP_evaluate  stdAP_evaluate
0  0.346456      0.187869        0.159078
1  0.406542      0.375058        0.101954
2  0.515274      0.473296        0.148758
3  0.467113      0.409678        0.143174


Patch angle 10 used with Faster RCNN Model
   mean_iou  mAP_evaluate  stdAP_evaluate
```

```
0   0.347118      0.185058        0.155806
1   0.407067      0.375604        0.102478
2   0.524567      0.482330        0.136606
3   0.431691      0.384603        0.134572


Patch angle 20 used with Faster RCNN Model
   mean_iou  mAP_evaluate  stdAP_evaluate
0   0.347143      0.188194        0.159249
1   0.407091      0.373749        0.089906
2   0.586845      0.533661        0.179801
3   0.439594      0.385666        0.132060


Patch angle 30 used with Faster RCNN Model
   mean_iou  mAP_evaluate  stdAP_evaluate
0   0.349138      0.189182        0.154313
1   0.406911      0.373890        0.091144
2   0.612773      0.555094        0.188835
3   0.475875      0.411346        0.149000


Patch angle 40 used with Faster RCNN Model
   mean_iou  mAP_evaluate  stdAP_evaluate
0   0.348364      0.187711        0.156188
1   0.405573      0.376931        0.101144
2   0.550552      0.505316        0.178019
3   0.459774      0.403933        0.144318
```

Out[60]:
```
Text(0.5, 1.0, 'Performance of Faster RCNN Model with Different Patch Angles')
```



In [ ]: