Tanawin-st123975-compare-models

November 23, 2023

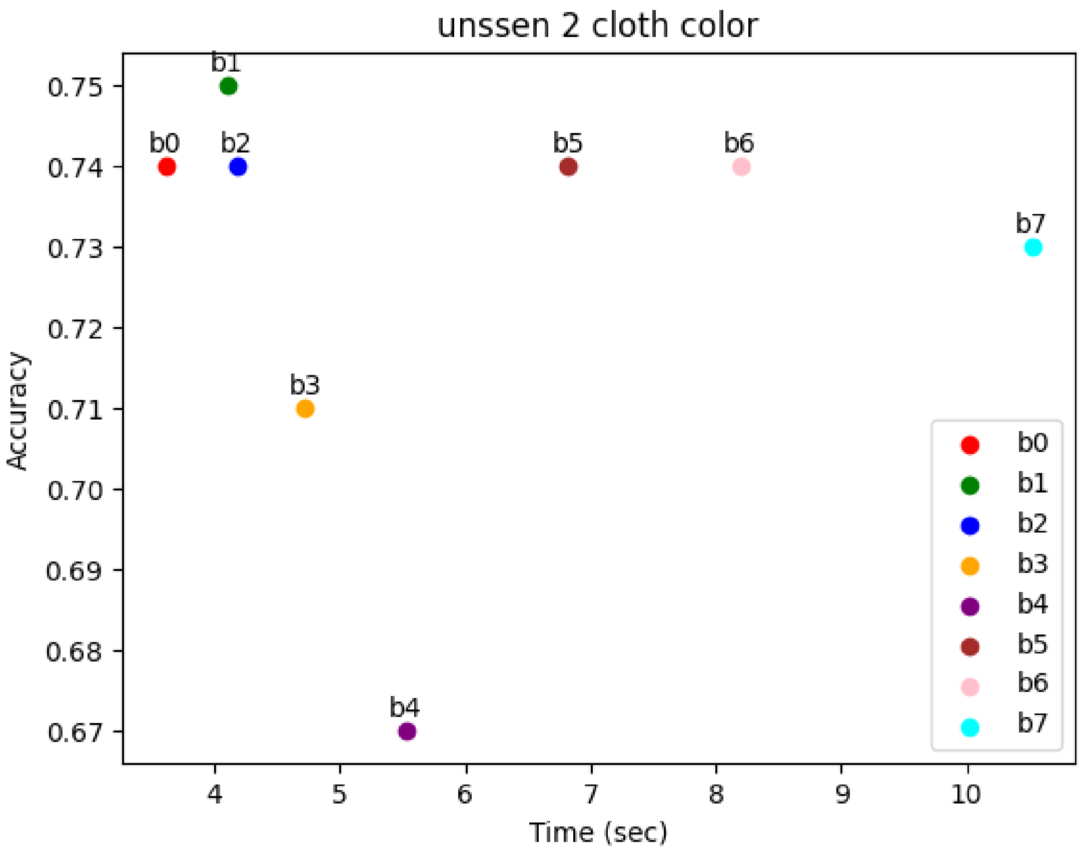
[ ]:
```
import matplotlib.pyplot as plt
```

1 plot function

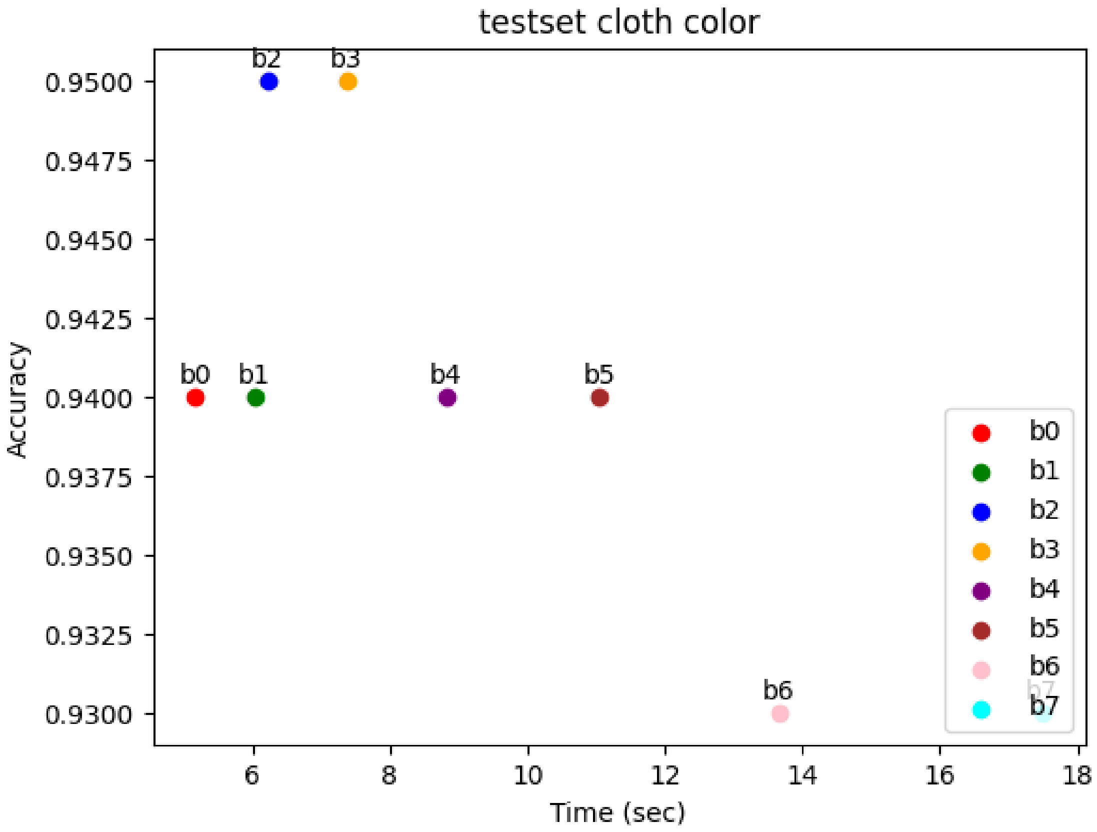
[ ]:
```
def compare_plot(x, y, dataset):
    eff = ["b0", "b1", "b2", "b3", "b4", "b5", "b6", "b7"]

    colors = ['red', 'green', 'blue', 'orange', 'purple', 'brown', 'pink',␣
 ↪'cyan']

    # Create scatter plot
    for i in range(len(eff)):
        plt.scatter(x[i], y[i], color=colors[i], label=eff[i])

    # Annotate each point with its corresponding label from the 'eff' list
    for i, txt in enumerate(eff):
        plt.annotate(txt, (x[i], y[i]), textcoords="offset points", xytext=(0,␣
 ↪5), ha='center')

    # Add labels and title
    plt.xlabel('Time (sec)')
    plt.ylabel('Accuracy')
    plt.title(dataset)

    # Add legend
    plt.legend(loc='lower right')
    # Show the plot
    plt.show()
```

2 Cloth Color Perframance

[ ]:
```
def unseen_2_color():
    time = [3.61, 4.1, 4.18, 4.72, 5.53, 6.82, 8.19, 10.52]
    acc = [0.74, 0.75, 0.74, 0.71, 0.67, 0.74, 0.74, 0.73]
    compare_plot(time, acc, "unssen 2 cloth color")

unseen_2_color()
```

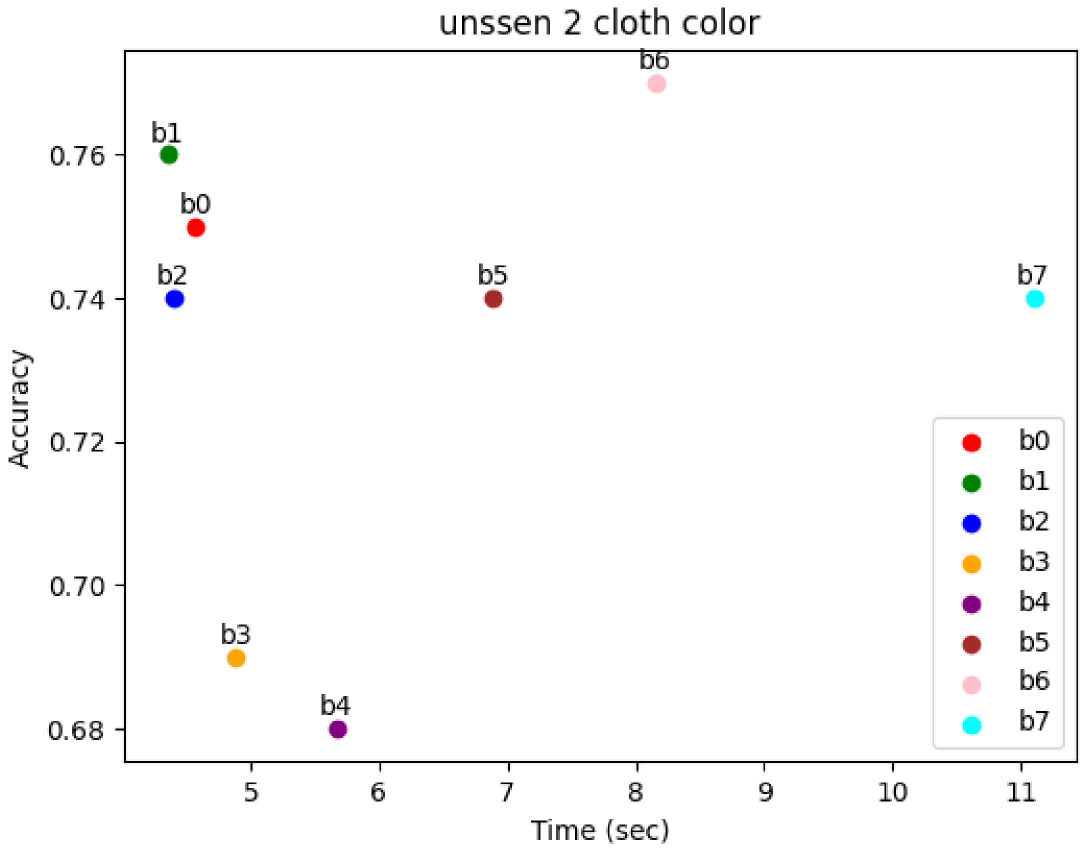


[ ]:
```
def testset_color():
    x_values = [5.16, 6.02, 6.22, 7.36, 8.81, 11.05, 13.65, 17.49]
    y_values = [0.94, 0.94, 0.95, 0.95, 0.94, 0.94, 0.93, 0.93]

    compare_plot(x_values, y_values, "testset cloth color")

testset_color()
```



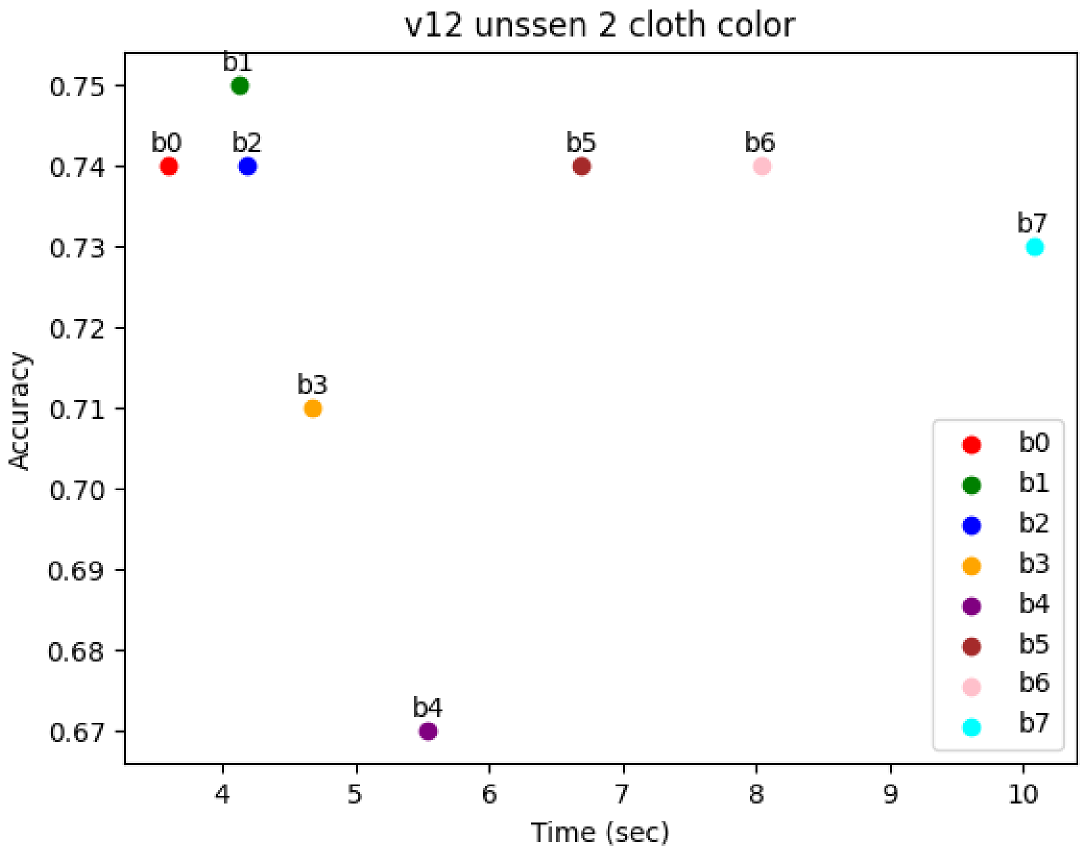
[ ]:

```python
def v_2_unseen_2_color():
    x1 = [4.57, 4.35, 4.4, 4.88, 5.67, 6.89, 8.15, 11.1]
    x2 = [0.75, 0.76, 0.74, 0.69, 0.68, 0.74, 0.77, 0.74]

    compare_plot(x1, x2, "unssen 2 cloth color")

v_2_unseen_2_color()
```



[ ]:

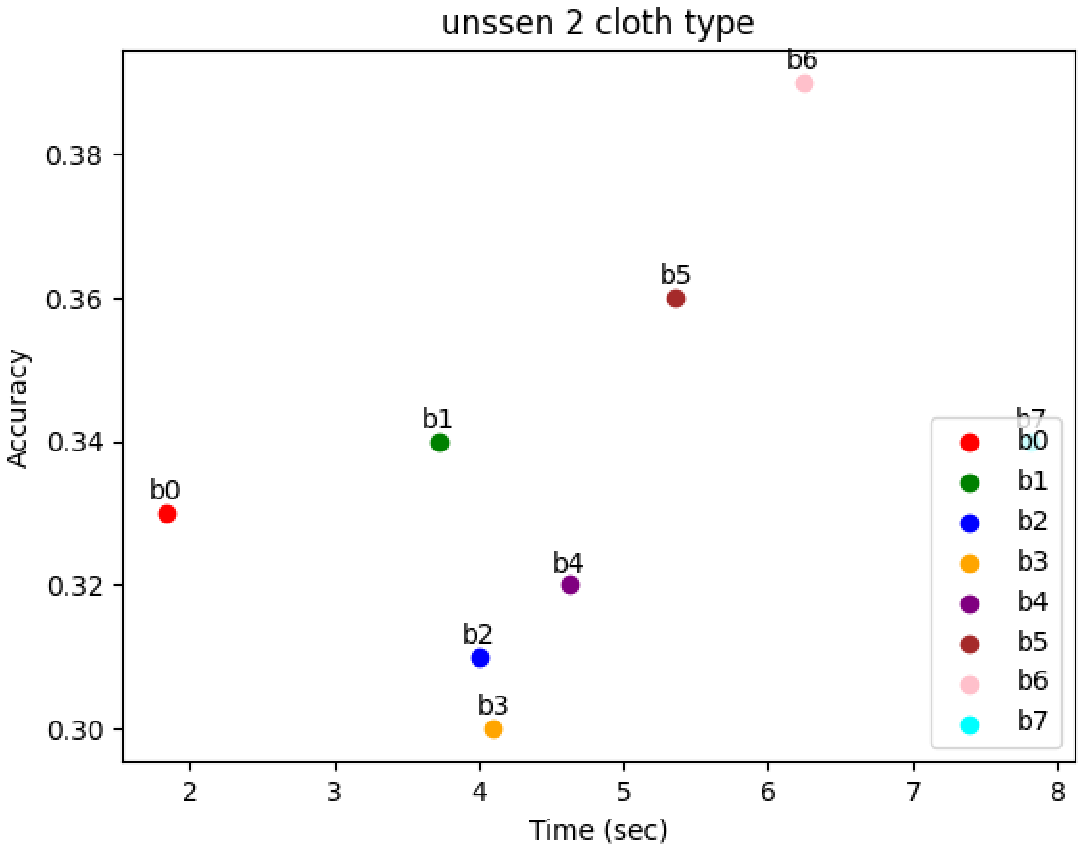
```
def v_12_unseen_2_color():
    x1 = [3.59, 4.12, 4.19, 4.68, 5.54, 6.69, 8.04, 10.08]
    x2 = [0.74, 0.75, 0.74, 0.71, 0.67, 0.74, 0.74, 0.73]

    compare_plot(x1, x2, "v12 unssen 2 cloth color")

v_12_unseen_2_color()
```



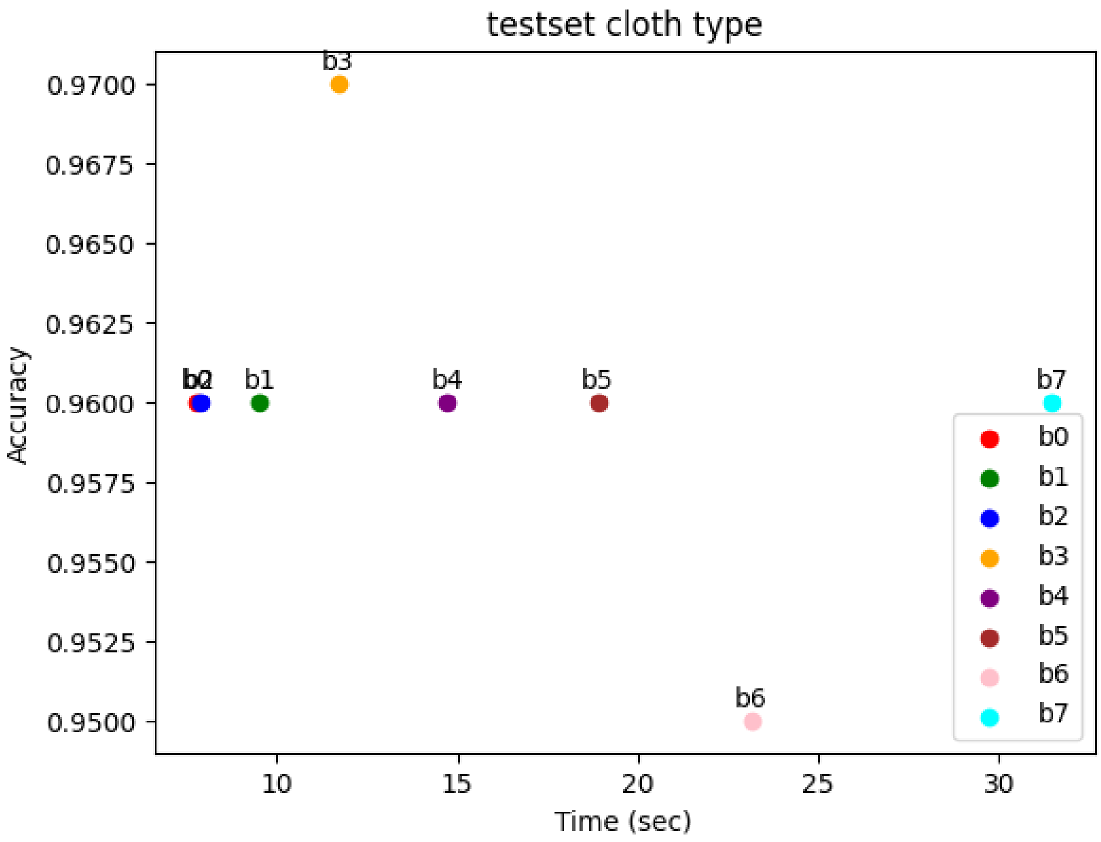
3 Cloth Type Perframance

[ ]:

```
def unseen_2_type():
    x_values = [1.83, 3.72, 4, 4.1, 4.62, 5.36, 6.24, 7.82]
    y_values = [0.33, 0.34, 0.31, 0.3, 0.32, 0.36, 0.39, 0.34]

    compare_plot(x_values, y_values, "unssen 2 cloth type")

unseen_2_type()
```

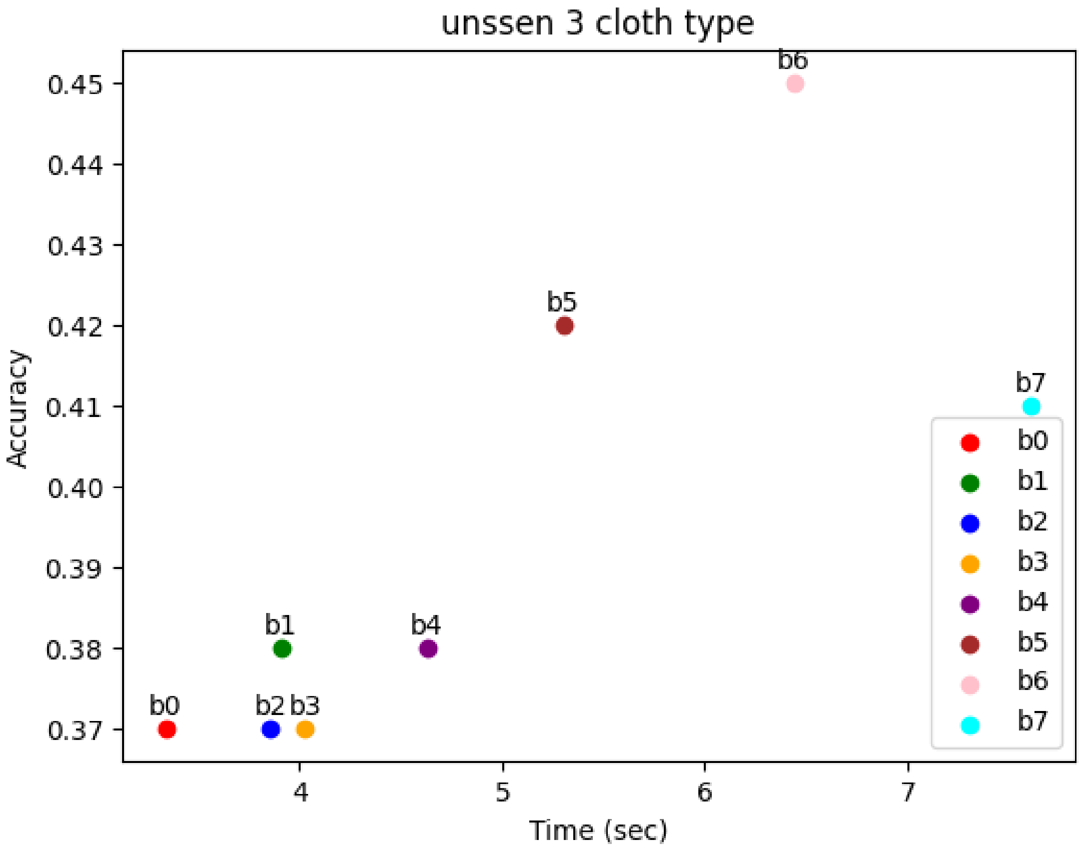


```
[ ]: def testset_type():
         x_values = [7.78, 9.52, 7.86, 11.69, 14.71, 18.88, 23.14, 31.46]
         y_values = [0.96, 0.96, 0.96, 0.97, 0.96, 0.96, 0.95, 0.96]

         compare_plot(x_values, y_values, "testset cloth type")

     testset_type()
```



[ ]:

```
def unseen_3_type():
    x_values = [3.34, 3.91, 3.86, 4.03, 4.63, 5.3, 6.44, 7.61]
    y_values = [0.37, 0.38, 0.37, 0.37, 0.38, 0.42, 0.45, 0.41]

    compare_plot(x_values, y_values, "unssen 3 cloth type")

unseen_3_type()
```



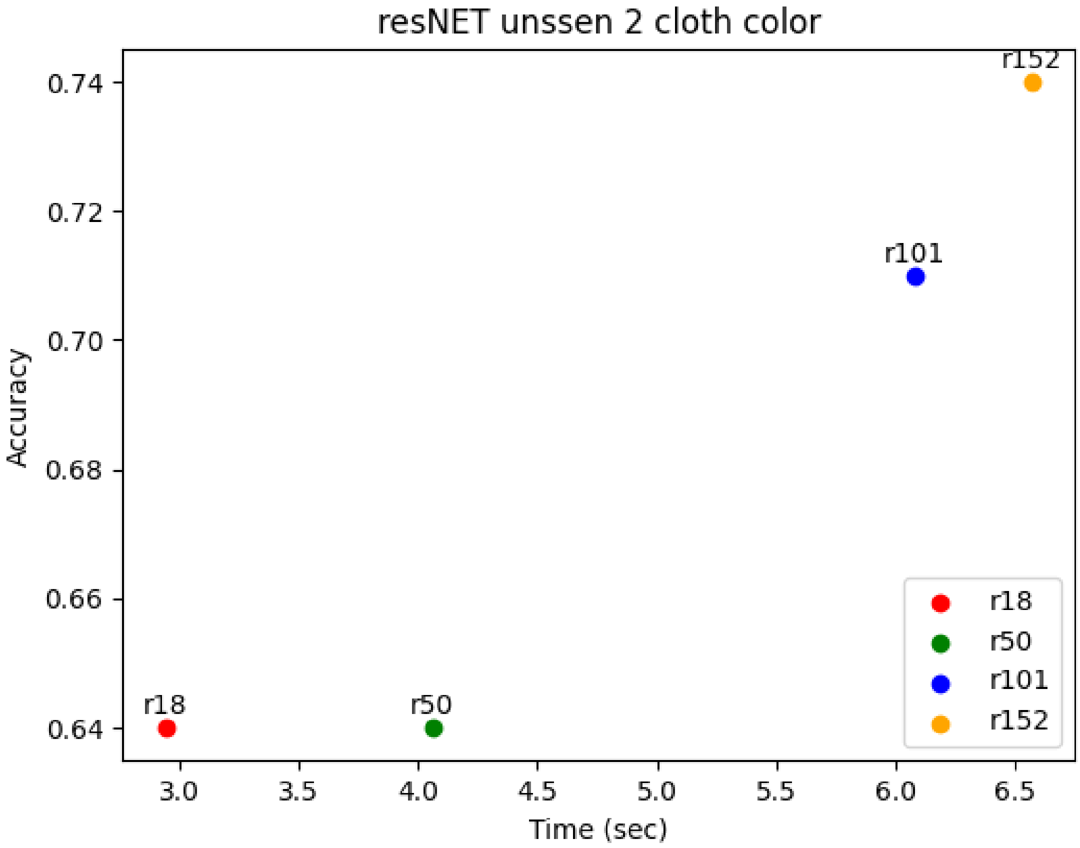
4 resNET color

```
[ ]: def res_compare_plot(x, y, dataset):
         eff = ["r18", "r50", "r101", "r152"]

         colors = ['red', 'green', 'blue', 'orange']

         # Create scatter plot
         for i in range(len(eff)):
             plt.scatter(x[i], y[i], color=colors[i], label=eff[i])

         # Annotate each point with its corresponding label from the 'eff' list
         for i, txt in enumerate(eff):
             plt.annotate(txt, (x[i], y[i]), textcoords="offset points", xytext=(0, 5), ha='center')

         # Add labels and title
         plt.xlabel('Time (sec)')
         plt.ylabel('Accuracy')
```

```
    plt.title(dataset)

    # Add legend
    plt.legend(loc='lower right')
    # Show the plot
    plt.show()
```

[ ]:

```
def res_unseen_2_color():
    time = [2.94, 4.06, 6.08, 6.57]
    acc = [0.64, 0.64, 0.71, 0.74]

    res_compare_plot(time, acc, "resNET unssen 2 cloth color")

res_unseen_2_color()
```

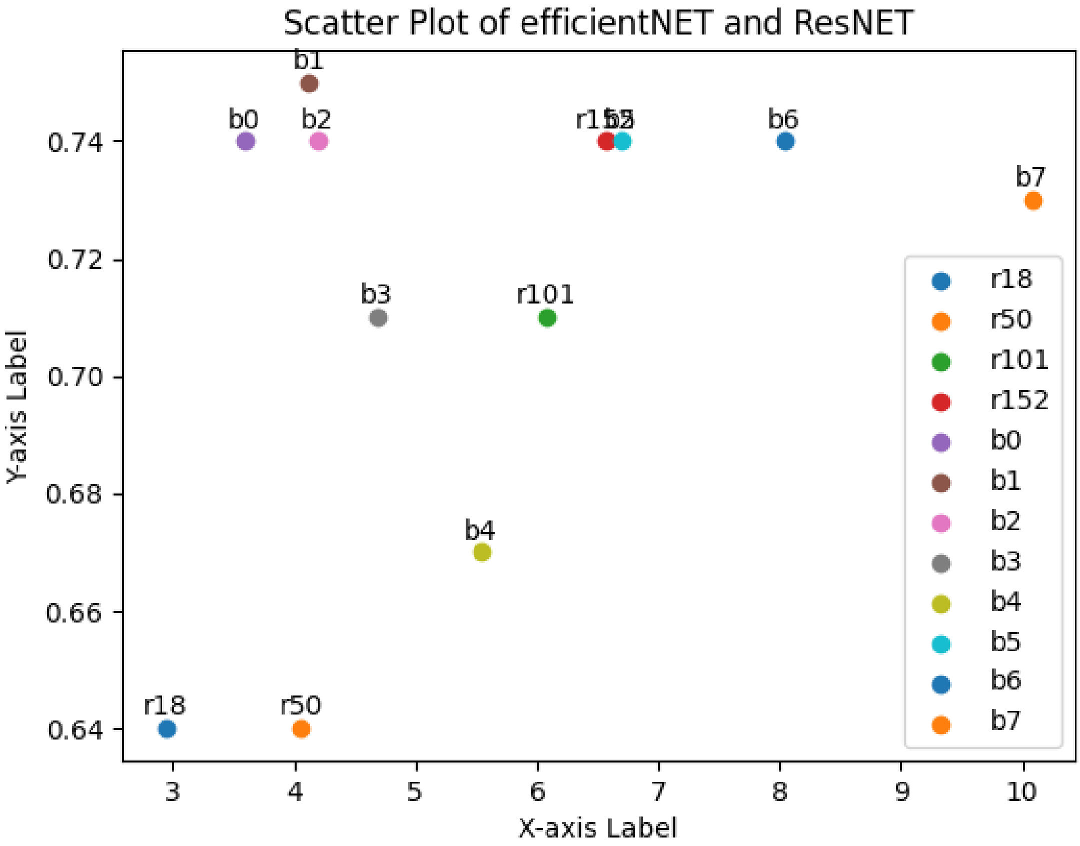


5 EfficientNET vs ResNET COLOR Unseen2

[ ]:

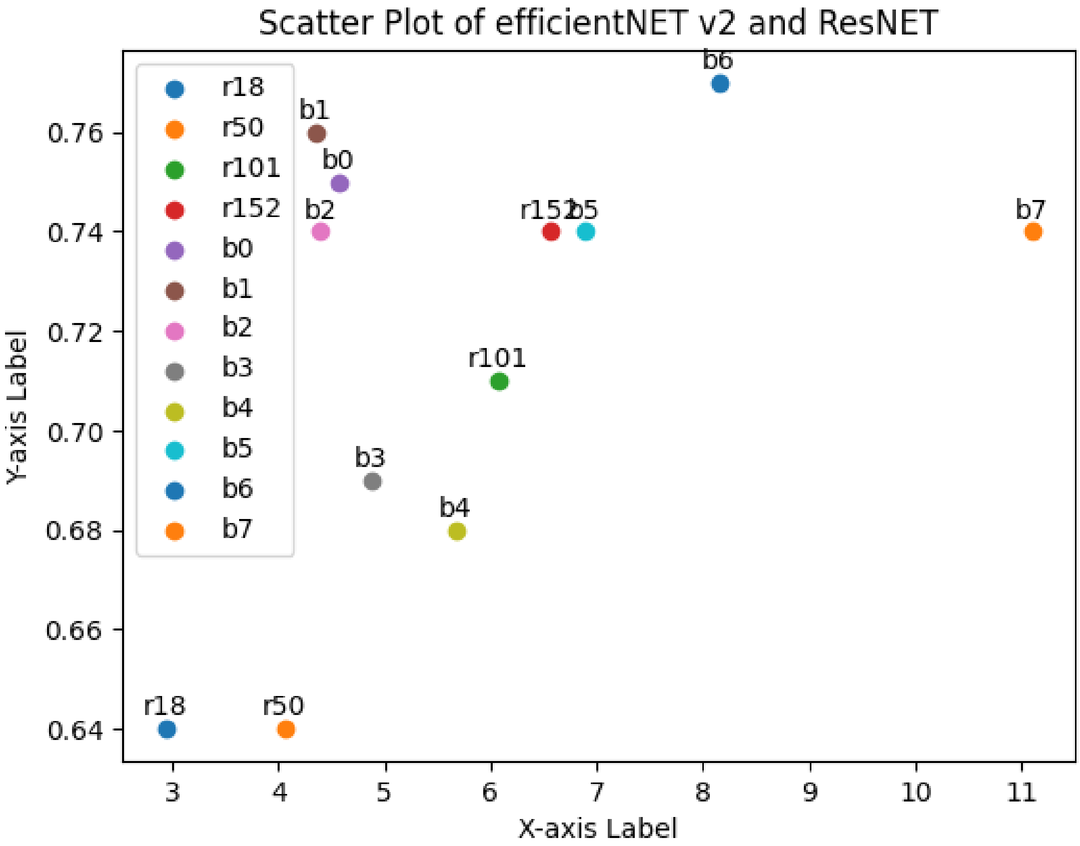
```
import matplotlib.pyplot as plt

# Data

    # x1 = [4.57, 4.35, 4.4, 4.88, 5.67, 6.89, 8.15, 11.1]
    # x2 = [0.75, 0.76, 0.74, 0.69, 0.68, 0.74, 0.77, 0.74]
eff_time = [3.59, 4.12, 4.19, 4.68, 5.54, 6.69, 8.04, 10.08]
eff_acc = [0.74, 0.75, 0.74, 0.71, 0.67, 0.74, 0.74, 0.73]

# eff_time = [4.57, 4.35, 4.4, 4.88, 5.67, 6.89, 8.15, 11.1]
# eff_acc = [0.75, 0.76, 0.74, 0.69, 0.68, 0.74, 0.77, 0.74]

res_time = [2.94, 4.06, 6.08, 6.57]
res_acc = [0.64, 0.64, 0.71, 0.74]

res = ["r18", "r50", "r101", "r152"]

for i in range(len(res)):
    plt.scatter(res_time[i], res_acc[i], label=res[i])

for i, txt in enumerate(res):
    plt.annotate(txt, (res_time[i], res_acc[i]), textcoords="offset points",␣
 ↪xytext=(0, 5), ha='center')

eff = ["b0", "b1", "b2", "b3", "b4", "b5", "b6", "b7"]


# Create scatter plot
for i in range(len(eff)):
        plt.scatter(eff_time[i], eff_acc[i], label=eff[i])

for i, txt in enumerate(eff):
    plt.annotate(txt, (eff_time[i], eff_acc[i]), textcoords="offset points",␣
 ↪xytext=(0, 5), ha='center')




# Adding labels and title
plt.xlabel('X-axis Label')
plt.ylabel('Y-axis Label')
plt.title('Scatter Plot of efficientNET and ResNET')

# Adding a legend
plt.legend()
```

```
# Display the plot
plt.show()
```



```
[ ]: import matplotlib.pyplot as plt

     # Data

         # x1 = [4.57, 4.35, 4.4, 4.88, 5.67, 6.89, 8.15, 11.1]
         # x2 = [0.75, 0.76, 0.74, 0.69, 0.68, 0.74, 0.77, 0.74]
     # eff_time = [3.59, 4.12, 4.19, 4.68, 5.54, 6.69, 8.04, 10.08]
     # eff_acc = [0.74, 0.75, 0.74, 0.71, 0.67, 0.74, 0.74, 0.73]

     eff_time = [4.57, 4.35, 4.4, 4.88, 5.67, 6.89, 8.15, 11.1]
     eff_acc = [0.75, 0.76, 0.74, 0.69, 0.68, 0.74, 0.77, 0.74]

     res_time = [2.94, 4.06, 6.08, 6.57]
     res_acc = [0.64, 0.64, 0.71, 0.74]

     res = ["r18", "r50", "r101", "r152"]
```

```
for i in range(len(res)):
    plt.scatter(res_time[i], res_acc[i], label=res[i])

for i, txt in enumerate(res):
    plt.annotate(txt, (res_time[i], res_acc[i]), textcoords="offset points", xytext=(0, 5), ha='center')

eff = ["b0", "b1", "b2", "b3", "b4", "b5", "b6", "b7"]


# Create scatter plot
for i in range(len(eff)):
        plt.scatter(eff_time[i], eff_acc[i], label=eff[i])

for i, txt in enumerate(eff):
    plt.annotate(txt, (eff_time[i], eff_acc[i]), textcoords="offset points", xytext=(0, 5), ha='center')



# Adding labels and title
plt.xlabel('X-axis Label')
plt.ylabel('Y-axis Label')
plt.title('Scatter Plot of efficientNET v2 and ResNET')

# Adding a legend
plt.legend()

# Display the plot
plt.show()
```

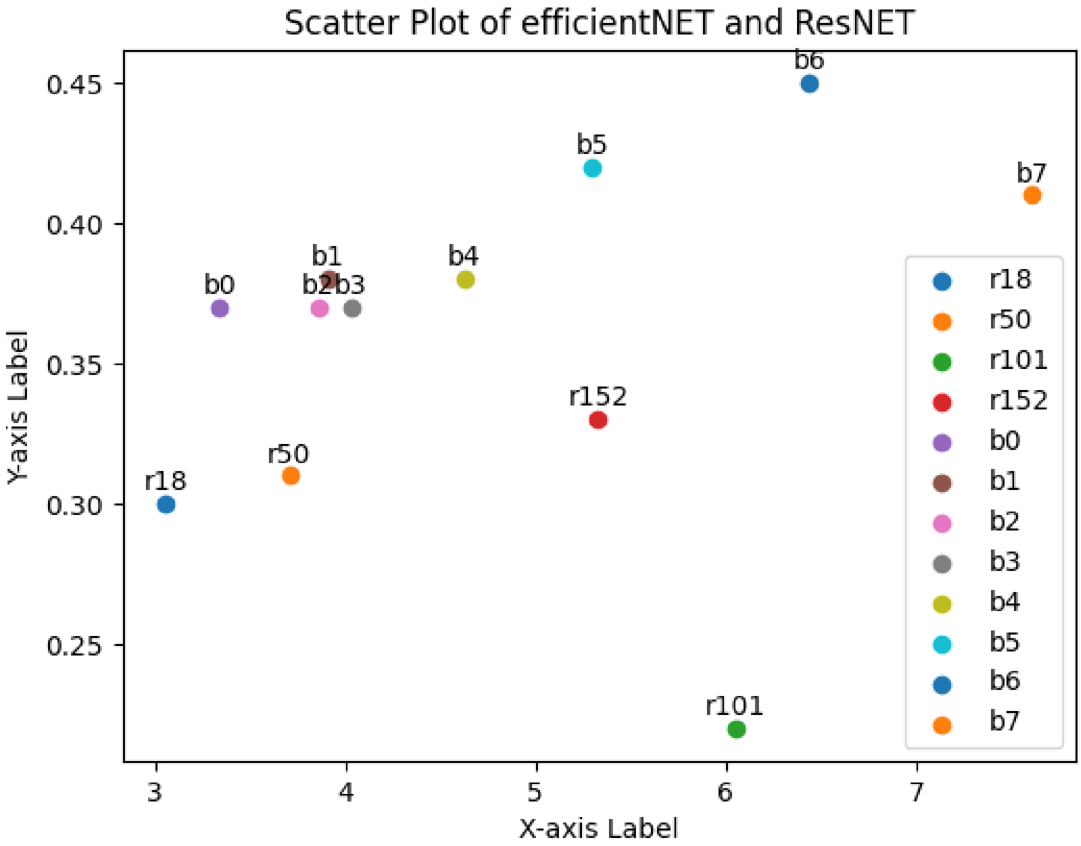


6 EfficientNET vs ResNET TYPE Unseen3

[ ]:

```
import matplotlib.pyplot as plt

# Data

    # x1 = [4.57, 4.35, 4.4, 4.88, 5.67, 6.89, 8.15, 11.1]
    # x2 = [0.75, 0.76, 0.74, 0.69, 0.68, 0.74, 0.77, 0.74]
# eff_time = [3.59, 4.12, 4.19, 4.68, 5.54, 6.69, 8.04, 10.08]
# eff_acc = [0.74, 0.75, 0.74, 0.71, 0.67, 0.74, 0.74, 0.73]

    # x_values = [3.34, 3.91, 3.86, 4.03, 4.63, 5.3, 6.44, 7.61]
    # y_values = [0.37, 0.38, 0.37, 0.37, 0.38, 0.42, 0.45, 0.41]

eff_time = [3.34, 3.91, 3.86, 4.03, 4.63, 5.3, 6.44, 7.61]
eff_acc = [0.37, 0.38, 0.37, 0.37, 0.38, 0.42, 0.45, 0.41]

# x1 = [3.06, 3.71, 6.05, 5.33]
# x2 = [0.3, 0.31, 0.22, 0.33]
```

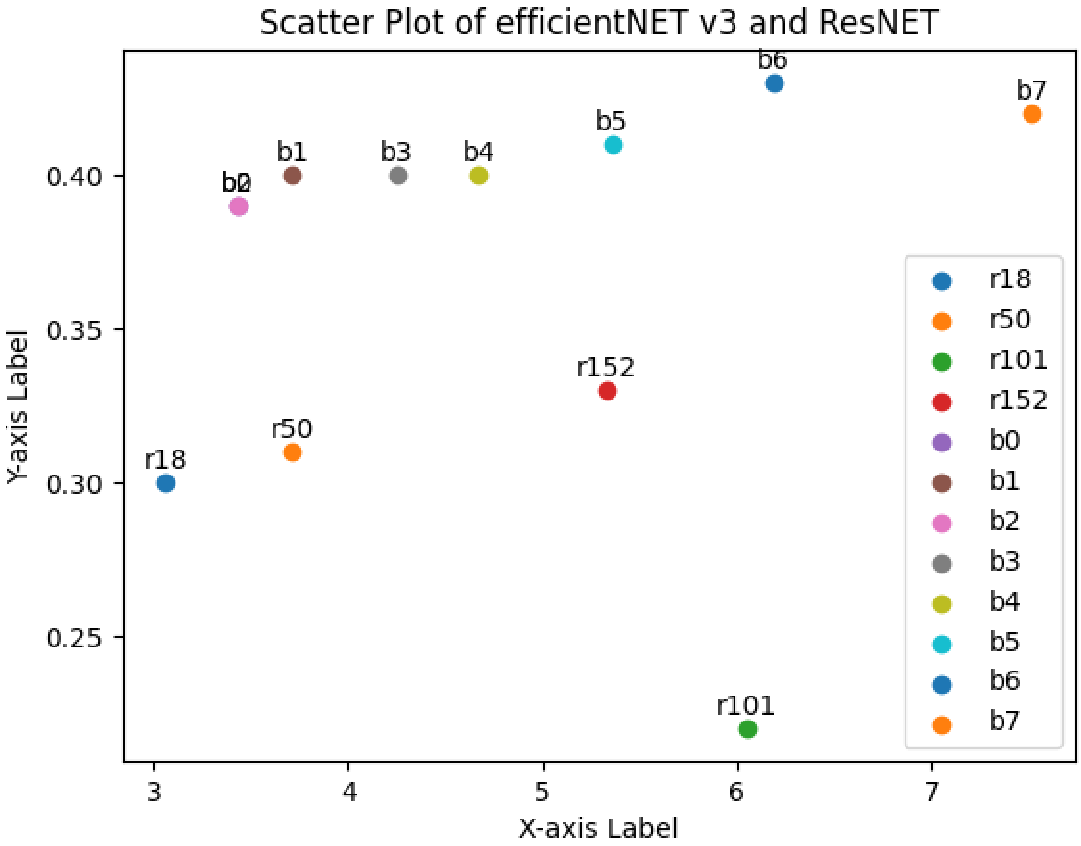
```
res_time = [3.06, 3.71, 6.05, 5.33]
res_acc = [0.3, 0.31, 0.22, 0.33]

res = ["r18", "r50", "r101", "r152"]

for i in range(len(res)):
    plt.scatter(res_time[i], res_acc[i], label=res[i])

for i, txt in enumerate(res):
    plt.annotate(txt, (res_time[i], res_acc[i]), textcoords="offset points",␣
 ↪xytext=(0, 5), ha='center')

eff = ["b0", "b1", "b2", "b3", "b4", "b5", "b6", "b7"]


# Create scatter plot
for i in range(len(eff)):
        plt.scatter(eff_time[i], eff_acc[i], label=eff[i])

for i, txt in enumerate(eff):
    plt.annotate(txt, (eff_time[i], eff_acc[i]), textcoords="offset points",␣
 ↪xytext=(0, 5), ha='center')



# Adding labels and title
plt.xlabel('X-axis Label')
plt.ylabel('Y-axis Label')
plt.title('Scatter Plot of efficientNET and ResNET')

# Adding a legend
plt.legend()

# Display the plot
plt.show()
```



[ ]:

```
import matplotlib.pyplot as plt

# Data

# x1 = [3.43, 3.71, 3.43, 4.25, 4.67, 5.36, 6.19, 7.52]
# x2 = [0.39, 0.4, 0.39, 0.4, 0.4, 0.41, 0.43, 0.42]

eff_time = [3.43, 3.71, 3.43, 4.25, 4.67, 5.36, 6.19, 7.52]
eff_acc = [0.39, 0.4, 0.39, 0.4, 0.4, 0.41, 0.43, 0.42]

# x1 = [3.06, 3.71, 6.05, 5.33]
# x2 = [0.3, 0.31, 0.22, 0.33]

res_time = [3.06, 3.71, 6.05, 5.33]
res_acc = [0.3, 0.31, 0.22, 0.33]

res = ["r18", "r50", "r101", "r152"]

for i in range(len(res)):
    plt.scatter(res_time[i], res_acc[i], label=res[i])
```

```python
for i, txt in enumerate(res):
    plt.annotate(txt, (res_time[i], res_acc[i]), textcoords="offset points", xytext=(0, 5), ha='center')

eff = ["b0", "b1", "b2", "b3", "b4", "b5", "b6", "b7"]


# Create scatter plot
for i in range(len(eff)):
        plt.scatter(eff_time[i], eff_acc[i], label=eff[i])

for i, txt in enumerate(eff):
    plt.annotate(txt, (eff_time[i], eff_acc[i]), textcoords="offset points", xytext=(0, 5), ha='center')



# Adding labels and title
plt.xlabel('X-axis Label')
plt.ylabel('Y-axis Label')
plt.title('Scatter Plot of efficientNET v3 and ResNET')

# Adding a legend
plt.legend()

# Display the plot
plt.show()
```


Scatter Plot of efficientNET v3 and ResNET
b6
b7
b5
b1
b3
b4
b0
b2
r152
r50
r18
r101
0.40
0.35
0.30
0.25
Y-axis Label
3
4
5
6
7
X-axis Label
r18
r50
r101
r152
b0
b1
b2
b3
b4
b5
b6
b7