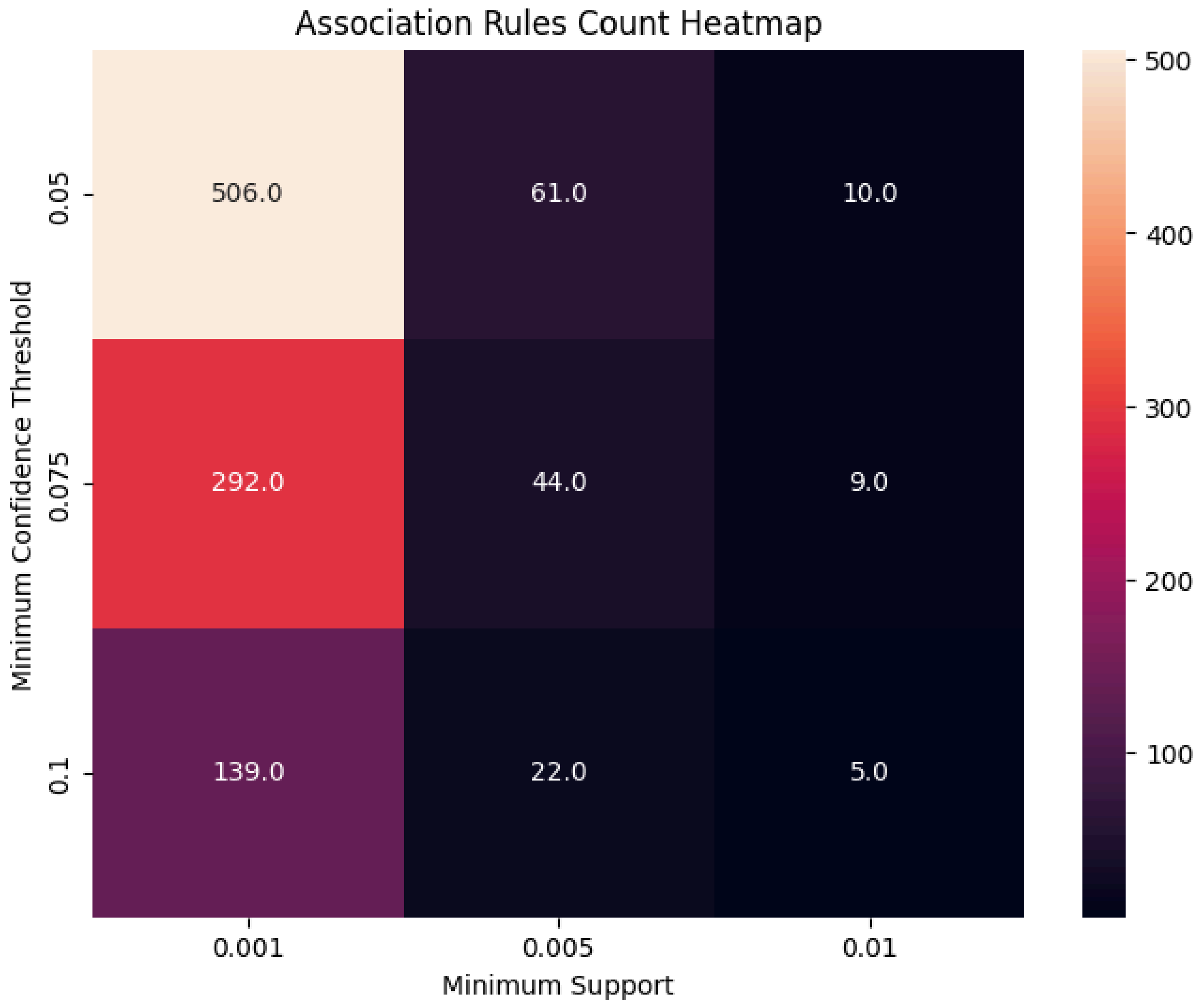
In [66]:
```
import cv2
import os
import numpy as np
import pandas as pd
import seaborn as sns
from glob import glob
import tensorflow as tf
import matplotlib.pyplot as plt
from tensorflow.keras.models import Sequential
from tensorflow.keras.utils import to_categorical
from sklearn.model_selection import train_test_split
from mlxtend.preprocessing import TransactionEncoder
from mlxtend.frequent_patterns import apriori,association_rules
from tensorflow.keras.preprocessing.image import ImageDataGenerator
import warnings
warnings.filterwarnings("ignore")
from keras.callbacks import Callback
from PIL import Image
import matplotlib.pyplot as plt
from tensorflow.keras.callbacks import Callback
from tensorflow.keras.layers import Conv2D, Dense, Flatten, MaxPooling2D
from tqdm import tqdm
from tqdm.notebook import tqdm
groceries= pd.read_csv(r"C:\Users\mohan\Downloads\Grocery_Items_7.csv")
grocery_list= [row.dropna().tolist() for index, row in groceries.iterrows()]
te = TransactionEncoder()
te_ary = te.fit(grocery_list).transform(grocery_list)
data = pd.DataFrame(te_ary, columns=te.columns_)

frequent_itemsets = apriori(data, min_support=0.01, use_colnames=True)
ar=association_rules(frequent_itemsets, metric="confidence", min_threshold=0.1)
print("\n  minimum support = 0.01 and minimum confidence threshold = 0.1, the association  rules generated : \n")
print(ar)

msv =[ 0.001, 0.005, 0.01]
mct =[ 0.05, 0.075, 0.1]
rows = []
for i in msv:
    items = apriori(data, min_support=i, use_colnames=True)
```

```
    for j in mct:
        ar = association_rules(items, metric="confidence", min_threshold=j)
        # Append row to the list
        rows.append({'msv': i, 'mct': j, 'count': len(ar)})
dataset = pd.DataFrame(rows)
glue = dataset.pivot(index='mct', columns='msv', values='count')
plt.figure(figsize=(8, 6))
sns.heatmap(glue, annot=True, fmt=".1f")
plt.title("Association Rules Count Heatmap")
plt.xlabel("Minimum Support")
plt.ylabel("Minimum Confidence Threshold")
plt.show()

s1 = data.iloc[:len(data)//2]
s2 = data.iloc[len(data)//2:]
s1_frequent_itemsets = apriori(s1, min_support=0.005, use_colnames=True)
s2_frequent_itemsets = apriori(s2, min_support=0.005, use_colnames=True)
s1_association_rules = association_rules(s1_frequent_itemsets, metric="confidence", min_threshold=0.075)
s2_association_rules = association_rules(s2_frequent_itemsets, metric="confidence", min_threshold=0.075)
print("\nAssociation Rules for Subset 1:")
print(s1_association_rules)
print("\nAssociation Rules for Subset 2:")
print(s2_association_rules)
common_rules = pd.merge(s1_association_rules, s2_association_rules, on=['antecedents', 'consequents'])
print("\nCommon Association Rules:")
print(common_rules)


class TQDMProgressBar(Callback):
    def on_train_begin(self, logs=None):
        self.epochs = self.params["epochs"]
        self.tqdm_bar = tqdm(total=self.epochs, desc="Training Progress")

    def on_epoch_end(self, epoch, logs=None):
        self.tqdm_bar.update(1)

    def on_train_end(self, logs=None):
        self.tqdm_bar.close()

def load_and_process_data(dataset_dir):
    X, y = [], []
```

```
    class_folders = [
        "C:\\Users\\mohan\\Desktop\\Cropped\\n02091635-otterhound",
        "C:\\Users\\mohan\\Desktop\\Cropped\\n02097209-standard_schnauzer",
        "C:\\Users\\mohan\\Desktop\\Cropped\\n02099712-Labrador_retriever",
        "C:\\Users\\mohan\\Desktop\\Cropped\\n02112137-chow",
    ]
    for class_index, folder_name in tqdm(enumerate(class_folders), total=len(class_folders), desc="Loading Data"):
        folder_path = os.path.join(dataset_dir, folder_name)
        for filename in os.listdir(folder_path):
            # img = np.load(os.path.join(folder_path, filename))

            img = Image.open(os.path.join(folder_path, filename))

            X.append(img)
            y.append(class_index)
    X = np.array(X) / 255.0
    y = to_categorical(y, num_classes=4)
    if X.ndim == 3:
        X = np.expand_dims(X, axis=-1)
    return train_test_split(X, y, test_size=0.2, random_state=42, stratify=y)

def build_model(input_shape, filter_size=(3, 3)):
    model = Sequential([
        Conv2D(8, filter_size, activation="relu", input_shape=input_shape),
        MaxPooling2D(2),
        Flatten(),
        Dense(16, activation="relu"),
        Dense(4, activation="softmax"),
    ])
    model.compile(
        optimizer="adam",
        loss="categorical_crossentropy",
        metrics=["accuracy"]
    )
    return model

def train_and_evaluate_model(model, X_train, y_train, filter_size):
    tqdm_callback = TQDMProgressBar()
    history = model.fit(
        X_train,
        y_train,
```

```
        epochs=20,
        batch_size=32,
        validation_split=0.2,
        callbacks=[tqdm_callback],
        verbose=0  # Turn off the default Keras progress bar
    )
    plot_learning_curves(history, filter_size)
    evaluate_model_performance(model, X_train, y_train, filter_size)

def plot_learning_curves(history, filter_size):
    plt.figure(figsize=(8, 6))
    plt.plot(history.history["accuracy"], label="Training Accuracy")
    plt.plot(history.history["val_accuracy"], label="Validation Accuracy")
    plt.title(f"Learning Curves for Filter Size {filter_size}")
    plt.xlabel("Epochs")
    plt.ylabel("Accuracy")
    plt.legend()
    plt.show()

def evaluate_model_performance(model, X_train, y_train, filter_size):
    train_score = model.evaluate(X_train, y_train, verbose=0)
    print(f"Performance for Filter Size {filter_size}:")
    print(f"Training Loss: {train_score[0]:.4f}, Training Accuracy: {train_score[1]*100:.2f}%\n")

dataset_dir = r"C:\Users\mohan\Desktop\Cropped"
X_train, X_test, y_train, y_test = load_and_process_data(dataset_dir)
input_shape = X_train.shape[1:]
print(input_shape)

print(f"Training model with filter size {(3, 3)}...")
model = build_model(input_shape, (3, 3))
train_and_evaluate_model(model, X_train, y_train, (3, 3))

print(f"Training model with filter size {(5, 5)}...")
model = build_model(input_shape, (5, 5))
train_and_evaluate_model(model, X_train, y_train, (5, 5))

print(f"Training model with filter size {(7, 7)}...")
model = build_model(input_shape, (7, 7))
train_and_evaluate_model(model, X_train, y_train, (7, 7))
```

```
minimum support = 0.01 and minimum confidence threshold = 0.1, the association  rules generated :

          antecedents         consequents  antecedent support  \
0       (rolls/buns)  (other vegetables)            0.111000
1  (other vegetables)        (whole milk)            0.118250
2       (rolls/buns)        (whole milk)            0.111000
3             (soda)        (whole milk)            0.097125
4           (yogurt)        (whole milk)            0.087750

   consequent support   support  confidence      lift  leverage  conviction  \
0             0.11825  0.011250    0.101351  0.857094 -0.001876    0.981195
1             0.15800  0.014250    0.120507  0.762705 -0.004433    0.957370
2             0.15800  0.014375    0.129505  0.819649 -0.003163    0.967265
3             0.15800  0.012750    0.131274  0.830849 -0.002596    0.969236
4             0.15800  0.011500    0.131054  0.829457 -0.002365    0.968990

   zhangs_metric
0      -0.157931
1      -0.260818
2      -0.198402
3      -0.183999
4      -0.183931
```


Association Rules Count Heatmap
Minimum Confidence Threshold
0.05
0.075
0.1
506.0
61.0
10.0
292.0
44.0
9.0
139.0
22.0
5.0
0.001
0.005
0.01
Minimum Support
500
400
300
200
100

```
Association Rules for Subset 1:
             antecedents          consequents  antecedent support  \
0          (bottled beer)  (other vegetables)             0.04250   
1          (bottled beer)        (whole milk)             0.04250   
2         (bottled water)  (other vegetables)             0.05950   
3         (bottled water)        (whole milk)             0.05950   
4           (canned beer)        (whole milk)             0.05000   
5          (citrus fruit)        (whole milk)             0.04750   
6           (frankfurter)  (other vegetables)             0.03400   
7           (frankfurter)        (whole milk)             0.03400   
8     (frozen vegetables)  (other vegetables)             0.03075   
9            (newspapers)        (whole milk)             0.04175   
10            (pip fruit)  (other vegetables)             0.04700   
11     (other vegetables)        (rolls/buns)             0.12300   
12           (rolls/buns)  (other vegetables)             0.10850   
13      (root vegetables)  (other vegetables)             0.07125   
14              (sausage)  (other vegetables)             0.06275   
15        (shopping bags)  (other vegetables)             0.04925   
16                 (soda)  (other vegetables)             0.09800   
17           (whole milk)  (other vegetables)             0.15450   
18     (other vegetables)        (whole milk)             0.12300   
19               (yogurt)  (other vegetables)             0.08175   
20               (pastry)        (whole milk)             0.04850   
21            (pip fruit)        (rolls/buns)             0.04700   
22            (pip fruit)              (soda)             0.04700   
23            (pip fruit)        (whole milk)             0.04700   
24                 (pork)        (whole milk)             0.03775   
25      (root vegetables)        (rolls/buns)             0.07125   
26              (sausage)        (rolls/buns)             0.06275   
27        (shopping bags)        (rolls/buns)             0.04925   
28                 (soda)        (rolls/buns)             0.09800   
29           (rolls/buns)              (soda)             0.10850   
30       (tropical fruit)        (rolls/buns)             0.06925   
31           (whole milk)        (rolls/buns)             0.15450   
32           (rolls/buns)        (whole milk)             0.10850   
33               (yogurt)        (rolls/buns)             0.08175   
34      (root vegetables)              (soda)             0.07125   
35      (root vegetables)        (whole milk)             0.07125   
36              (sausage)              (soda)             0.06275   
37              (sausage)        (whole milk)             0.06275   
38        (shopping bags)              (soda)             0.04925   
```

```
39      (shopping bags)      (whole milk)            0.04925
40               (soda)      (whole milk)            0.09800
41         (whole milk)            (soda)            0.15450
42               (soda)          (yogurt)            0.09800
43             (yogurt)            (soda)            0.08175
44      (tropical fruit)     (whole milk)            0.06925
45  (whipped/sour cream)     (whole milk)            0.04825
46             (yogurt)      (whole milk)            0.08175

    consequent support  support  confidence      lift  leverage  conviction  \
0              0.12300  0.00500    0.117647  0.956480 -0.000228    0.993933
1              0.15450  0.00625    0.147059  0.951837 -0.000316    0.991276
2              0.12300  0.00525    0.088235  0.717360 -0.002068    0.961871
3              0.15450  0.00675    0.113445  0.734274 -0.002443    0.953692
4              0.15450  0.00550    0.110000  0.711974 -0.002225    0.950000
5              0.15450  0.00725    0.152632  0.987907 -0.000089    0.997795
6              0.12300  0.00500    0.147059  1.195600  0.000818    1.028207
7              0.15450  0.00550    0.161765  1.047021  0.000247    1.008667
8              0.12300  0.00500    0.162602  1.321964  0.001218    1.047291
9              0.15450  0.00500    0.119760  0.775149 -0.001450    0.960534
10             0.12300  0.00525    0.111702  0.908147 -0.000531    0.987281
11             0.10850  0.01275    0.103659  0.955378 -0.000596    0.994599
12             0.12300  0.01275    0.117512  0.955378 -0.000596    0.993781
13             0.12300  0.00600    0.084211  0.684638 -0.002764    0.957644
14             0.12300  0.00625    0.099602  0.809769 -0.001468    0.974013
15             0.12300  0.00525    0.106599  0.866658 -0.000808    0.981642
16             0.12300  0.00900    0.091837  0.746640 -0.003054    0.965685
17             0.12300  0.01775    0.114887  0.934038 -0.001254    0.990834
18             0.15450  0.01775    0.144309  0.934038 -0.001254    0.988090
19             0.12300  0.00700    0.085627  0.696154 -0.003055    0.959127
20             0.15450  0.00525    0.108247  0.700631 -0.002243    0.948133
21             0.10850  0.00500    0.106383  0.980488 -0.000099    0.997631
22             0.09800  0.00500    0.106383  1.085541  0.000394    1.009381
23             0.15450  0.00800    0.170213  1.101701  0.000739    1.018936
24             0.15450  0.00575    0.152318  0.985876 -0.000082    0.997426
25             0.10850  0.00600    0.084211  0.776134 -0.001731    0.973477
26             0.10850  0.00575    0.091633  0.844548 -0.001058    0.981432
27             0.10850  0.00575    0.116751  1.076049  0.000406    1.009342
28             0.10850  0.00950    0.096939  0.893445 -0.001133    0.987198
29             0.09800  0.00950    0.087558  0.893445 -0.001133    0.988556
30             0.10850  0.00750    0.108303  0.998187 -0.000014    0.999779
```

31	0.10850	0.01400	0.090615	0.835160	-0.002763	0.980333
32	0.15450	0.01400	0.129032	0.835160	-0.002763	0.970759
33	0.10850	0.00650	0.079511	0.732818	-0.002370	0.968507
34	0.09800	0.00550	0.077193	0.787683	-0.001483	0.977452
35	0.15450	0.00875	0.122807	0.794867	-0.002258	0.963870
36	0.09800	0.00650	0.103586	1.056997	0.000350	1.006231
37	0.15450	0.00950	0.151394	0.979899	-0.000195	0.996340
38	0.09800	0.00500	0.101523	1.035947	0.000174	1.003921
39	0.15450	0.00775	0.157360	1.018514	0.000141	1.003395
40	0.15450	0.01525	0.155612	1.007199	0.000109	1.001317
41	0.09800	0.01525	0.098706	1.007199	0.000109	1.000783
42	0.08175	0.00775	0.079082	0.967359	-0.000262	0.997102
43	0.09800	0.00775	0.094801	0.967359	-0.000262	0.996466
44	0.15450	0.01025	0.148014	0.958022	-0.000449	0.992388
45	0.15450	0.00500	0.103627	0.670725	-0.002455	0.943246
46	0.15450	0.01050	0.128440	0.831329	-0.002130	0.970100

	zhangs_metric
0	-0.045364
1	-0.050193
2	-0.295242
3	-0.277865
4	-0.298658
5	-0.012689
6	0.169358
7	0.046490
8	0.251277
9	-0.232371
10	-0.095948
11	-0.050564
12	-0.049782
13	-0.331534
14	-0.200415
15	-0.139287
16	-0.273362
17	-0.077086
18	-0.074523
19	-0.322182
20	-0.309900
21	-0.020454
22	0.082686

```
23       0.096865
24      -0.014670
25      -0.236970
26      -0.164151
27       0.074335
28      -0.116780
29      -0.117993
30      -0.001948
31      -0.189260
32      -0.181265
33      -0.284209
34      -0.224941
35      -0.217448
36       0.057533
37      -0.021418
38       0.036498
39       0.019119
40       0.007924
41       0.008454
42      -0.036059
43      -0.035444
44      -0.044961
45      -0.340288
46      -0.180970

Association Rules for Subset 2:
                antecedents         consequents  antecedent support  \
0             (bottled beer)  (other vegetables)             0.04525
1             (bottled beer)        (rolls/buns)             0.04525
2             (bottled beer)        (whole milk)             0.04525
3            (bottled water)  (other vegetables)             0.06300
4            (bottled water)        (rolls/buns)             0.06300
5            (bottled water)              (soda)             0.06300
6            (bottled water)        (whole milk)             0.06300
7              (brown bread)        (whole milk)             0.03525
8                   (butter)        (whole milk)             0.03325
9             (citrus fruit)  (other vegetables)             0.05675
10            (citrus fruit)        (whole milk)             0.05675
11           (domestic eggs)        (whole milk)             0.03625
12             (frankfurter)        (whole milk)             0.03725
13  (fruit/vegetable juice)        (whole milk)             0.03425
```

14	(newspapers)	(whole milk)	0.03625
15	(other vegetables)	(rolls/buns)	0.11350
16	(rolls/buns)	(other vegetables)	0.11350
17	(root vegetables)	(other vegetables)	0.06925
18	(soda)	(other vegetables)	0.09625
19	(other vegetables)	(soda)	0.11350
20	(tropical fruit)	(other vegetables)	0.06850
21	(whipped/sour cream)	(other vegetables)	0.04375
22	(other vegetables)	(whole milk)	0.11350
23	(other vegetables)	(yogurt)	0.11350
24	(yogurt)	(other vegetables)	0.09375
25	(pastry)	(whole milk)	0.05325
26	(pip fruit)	(whole milk)	0.04550
27	(pork)	(whole milk)	0.03350
28	(root vegetables)	(rolls/buns)	0.06925
29	(sausage)	(rolls/buns)	0.05950
30	(soda)	(rolls/buns)	0.09625
31	(tropical fruit)	(rolls/buns)	0.06850
32	(whole milk)	(rolls/buns)	0.16150
33	(rolls/buns)	(whole milk)	0.11350
34	(rolls/buns)	(yogurt)	0.11350
35	(yogurt)	(rolls/buns)	0.09375
36	(root vegetables)	(whole milk)	0.06925
37	(root vegetables)	(yogurt)	0.06925
38	(sausage)	(whole milk)	0.05950
39	(sausage)	(yogurt)	0.05950
40	(shopping bags)	(whole milk)	0.04575
41	(tropical fruit)	(soda)	0.06850
42	(soda)	(whole milk)	0.09625
43	(tropical fruit)	(whole milk)	0.06850
44	(tropical fruit)	(yogurt)	0.06850
45	(whipped/sour cream)	(whole milk)	0.04375
46	(whole milk)	(yogurt)	0.16150
47	(yogurt)	(whole milk)	0.09375

	consequent support	support	confidence	lift	leverage	conviction	\
0	0.11350	0.00625	0.138122	1.216930	0.001114	1.028567	
1	0.11350	0.00500	0.110497	0.973544	-0.000136	0.996624	
2	0.16150	0.00775	0.171271	1.060500	0.000442	1.011790	
3	0.11350	0.00650	0.103175	0.909027	-0.000651	0.988487	
4	0.11350	0.00600	0.095238	0.839102	-0.001151	0.979816	

5	0.09625	0.00600	0.095238	0.989487	-0.000064	0.998882
6	0.16150	0.00675	0.107143	0.663423	-0.003424	0.939120
7	0.16150	0.00575	0.163121	1.010034	0.000057	1.001936
8	0.16150	0.00500	0.150376	0.931120	-0.000370	0.986907
9	0.11350	0.00525	0.092511	0.815075	-0.001191	0.976871
10	0.16150	0.00725	0.127753	0.791042	-0.001915	0.961311
11	0.16150	0.00600	0.165517	1.024875	0.000146	1.004814
12	0.16150	0.00525	0.140940	0.872691	-0.000766	0.976066
13	0.16150	0.00650	0.189781	1.175115	0.000969	1.034905
14	0.16150	0.00650	0.179310	1.110281	0.000646	1.021702
15	0.11350	0.00975	0.085903	0.756855	-0.003132	0.969810
16	0.11350	0.00975	0.085903	0.756855	-0.003132	0.969810
17	0.11350	0.00550	0.079422	0.699757	-0.002360	0.962982
18	0.11350	0.00925	0.096104	0.846730	-0.001674	0.980754
19	0.09625	0.00925	0.081498	0.846730	-0.001674	0.983939
20	0.11350	0.00800	0.116788	1.028972	0.000225	1.003723
21	0.11350	0.00500	0.114286	1.006923	0.000034	1.000887
22	0.16150	0.01075	0.094714	0.586462	-0.007580	0.926226
23	0.09375	0.00950	0.083700	0.892805	-0.001141	0.989032
24	0.11350	0.00950	0.101333	0.892805	-0.001141	0.986461
25	0.16150	0.00750	0.140845	0.872106	-0.001100	0.975959
26	0.16150	0.00575	0.126374	0.782499	-0.001598	0.959792
27	0.16150	0.00500	0.149254	0.924172	-0.000410	0.985605
28	0.11350	0.00625	0.090253	0.795178	-0.001610	0.974446
29	0.11350	0.00600	0.100840	0.888461	-0.000753	0.985921
30	0.11350	0.00800	0.083117	0.732307	-0.002924	0.966863
31	0.11350	0.00625	0.091241	0.803884	-0.001525	0.975506
32	0.11350	0.01475	0.091331	0.804681	-0.003580	0.975603
33	0.16150	0.01475	0.129956	0.804681	-0.003580	0.963744
34	0.09375	0.00950	0.083700	0.892805	-0.001141	0.989032
35	0.11350	0.00950	0.101333	0.892805	-0.001141	0.986461
36	0.16150	0.00700	0.101083	0.625901	-0.004184	0.932789
37	0.09375	0.00550	0.079422	0.847172	-0.000992	0.984436
38	0.16150	0.00800	0.134454	0.832531	-0.001609	0.968752
39	0.09375	0.00650	0.109244	1.165266	0.000922	1.017394
40	0.16150	0.00575	0.125683	0.778223	-0.001639	0.959034
41	0.09625	0.00575	0.083942	0.872121	-0.000843	0.986564
42	0.16150	0.01025	0.106494	0.659403	-0.005294	0.938438
43	0.16150	0.00650	0.094891	0.587557	-0.004563	0.926407
44	0.09375	0.00625	0.091241	0.973236	-0.000172	0.997239
45	0.16150	0.00575	0.131429	0.813799	-0.001316	0.965378

```
46             0.09375  0.01250    0.077399  0.825593 -0.002641    0.982278
47             0.16150  0.01250    0.133333  0.825593 -0.002641    0.967500

    zhangs_metric
0        0.186709
1       -0.027675
2        0.059752
3       -0.096499
4       -0.169878
5       -0.011212
6       -0.351258
7        0.010298
8       -0.071080
9       -0.193894
10      -0.218779
11       0.025184
12      -0.131587
13       0.154304
14       0.103063
15      -0.265994
16      -0.265994
17      -0.315533
18      -0.166869
19      -0.169566
20       0.030227
21       0.007190
22      -0.443027
23      -0.119283
24      -0.116987
25      -0.134123
26      -0.225530
27      -0.078251
28      -0.216758
29      -0.117764
30      -0.287992
31      -0.207544
32      -0.224493
33      -0.214951
34      -0.119283
35      -0.116987
36      -0.391048
```

```
37      -0.162353
38      -0.176197
39       0.150799
40      -0.229964
41      -0.136004
42      -0.363679
43      -0.429739
44      -0.028676
45      -0.193075
46      -0.201238
47      -0.189038

Common Association Rules:
             antecedents          consequents  antecedent support_x  \
0          (bottled beer)  (other vegetables)               0.04250
1          (bottled beer)        (whole milk)               0.04250
2         (bottled water)  (other vegetables)               0.05950
3         (bottled water)        (whole milk)               0.05950
4          (citrus fruit)        (whole milk)               0.04750
5           (frankfurter)        (whole milk)               0.03400
6            (newspapers)        (whole milk)               0.04175
7     (other vegetables)        (rolls/buns)               0.12300
8            (rolls/buns)  (other vegetables)               0.10850
9      (root vegetables)  (other vegetables)               0.07125
10                 (soda)  (other vegetables)               0.09800
11    (other vegetables)        (whole milk)               0.12300
12               (yogurt)  (other vegetables)               0.08175
13               (pastry)        (whole milk)               0.04850
14            (pip fruit)        (whole milk)               0.04700
15                 (pork)        (whole milk)               0.03775
16     (root vegetables)        (rolls/buns)               0.07125
17              (sausage)        (rolls/buns)               0.06275
18                 (soda)        (rolls/buns)               0.09800
19      (tropical fruit)        (rolls/buns)               0.06925
20           (whole milk)        (rolls/buns)               0.15450
21           (rolls/buns)        (whole milk)               0.10850
22               (yogurt)        (rolls/buns)               0.08175
23     (root vegetables)        (whole milk)               0.07125
24              (sausage)        (whole milk)               0.06275
25        (shopping bags)        (whole milk)               0.04925
26                 (soda)        (whole milk)               0.09800
```

```
27     (tropical fruit)       (whole milk)              0.06925
28  (whipped/sour cream)       (whole milk)              0.04825
29              (yogurt)       (whole milk)              0.08175

    consequent support_x  support_x  confidence_x    lift_x  leverage_x  \
0                 0.1230    0.00500      0.117647  0.956480   -0.000228
1                 0.1545    0.00625      0.147059  0.951837   -0.000316
2                 0.1230    0.00525      0.088235  0.717360   -0.002068
3                 0.1545    0.00675      0.113445  0.734274   -0.002443
4                 0.1545    0.00725      0.152632  0.987907   -0.000089
5                 0.1545    0.00550      0.161765  1.047021    0.000247
6                 0.1545    0.00500      0.119760  0.775149   -0.001450
7                 0.1085    0.01275      0.103659  0.955378   -0.000596
8                 0.1230    0.01275      0.117512  0.955378   -0.000596
9                 0.1230    0.00600      0.084211  0.684638   -0.002764
10                0.1230    0.00900      0.091837  0.746640   -0.003054
11                0.1545    0.01775      0.144309  0.934038   -0.001254
12                0.1230    0.00700      0.085627  0.696154   -0.003055
13                0.1545    0.00525      0.108247  0.700631   -0.002243
14                0.1545    0.00800      0.170213  1.101701    0.000739
15                0.1545    0.00575      0.152318  0.985876   -0.000082
16                0.1085    0.00600      0.084211  0.776134   -0.001731
17                0.1085    0.00575      0.091633  0.844548   -0.001058
18                0.1085    0.00950      0.096939  0.893445   -0.001133
19                0.1085    0.00750      0.108303  0.998187   -0.000014
20                0.1085    0.01400      0.090615  0.835160   -0.002763
21                0.1545    0.01400      0.129032  0.835160   -0.002763
22                0.1085    0.00650      0.079511  0.732818   -0.002370
23                0.1545    0.00875      0.122807  0.794867   -0.002258
24                0.1545    0.00950      0.151394  0.979899   -0.000195
25                0.1545    0.00775      0.157360  1.018514    0.000141
26                0.1545    0.01525      0.155612  1.007199    0.000109
27                0.1545    0.01025      0.148014  0.958022   -0.000449
28                0.1545    0.00500      0.103627  0.670725   -0.002455
29                0.1545    0.01050      0.128440  0.831329   -0.002130

    conviction_x  zhangs_metric_x  antecedent support_y  consequent support_y  \
0       0.993933        -0.045364               0.04525                0.1135
1       0.991276        -0.050193               0.04525                0.1615
2       0.961871        -0.295242               0.06300                0.1135
3       0.953692        -0.277865               0.06300                0.1615
```

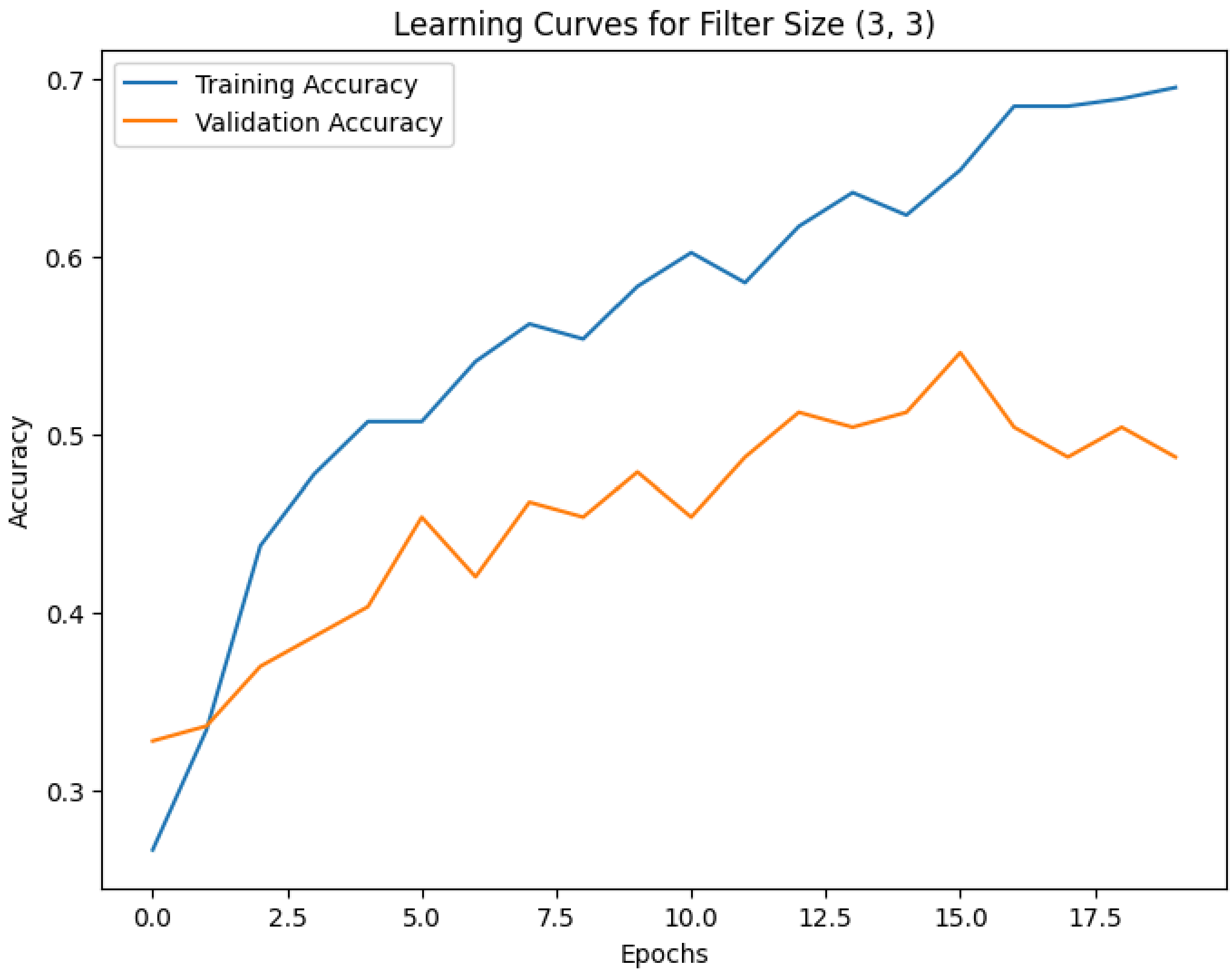
4	0.997795	-0.012689	0.05675	0.1615
5	1.008667	0.046490	0.03725	0.1615
6	0.960534	-0.232371	0.03625	0.1615
7	0.994599	-0.050564	0.11350	0.1135
8	0.993781	-0.049782	0.11350	0.1135
9	0.957644	-0.331534	0.06925	0.1135
10	0.965685	-0.273362	0.09625	0.1135
11	0.988090	-0.074523	0.11350	0.1615
12	0.959127	-0.322182	0.09375	0.1135
13	0.948133	-0.309900	0.05325	0.1615
14	1.018936	0.096865	0.04550	0.1615
15	0.997426	-0.014670	0.03350	0.1615
16	0.973477	-0.236970	0.06925	0.1135
17	0.981432	-0.164151	0.05950	0.1135
18	0.987198	-0.116780	0.09625	0.1135
19	0.999779	-0.001948	0.06850	0.1135
20	0.980333	-0.189260	0.16150	0.1135
21	0.970759	-0.181265	0.11350	0.1615
22	0.968507	-0.284209	0.09375	0.1135
23	0.963870	-0.217448	0.06925	0.1615
24	0.996340	-0.021418	0.05950	0.1615
25	1.003395	0.019119	0.04575	0.1615
26	1.001317	0.007924	0.09625	0.1615
27	0.992388	-0.044961	0.06850	0.1615
28	0.943246	-0.340288	0.04375	0.1615
29	0.970100	-0.180970	0.09375	0.1615

	support_y	confidence_y	lift_y	leverage_y	conviction_y	\
0	0.00625	0.138122	1.216930	0.001114	1.028567	
1	0.00775	0.171271	1.060500	0.000442	1.011790	
2	0.00650	0.103175	0.909027	-0.000651	0.988487	
3	0.00675	0.107143	0.663423	-0.003424	0.939120	
4	0.00725	0.127753	0.791042	-0.001915	0.961311	
5	0.00525	0.140940	0.872691	-0.000766	0.976066	
6	0.00650	0.179310	1.110281	0.000646	1.021702	
7	0.00975	0.085903	0.756855	-0.003132	0.969810	
8	0.00975	0.085903	0.756855	-0.003132	0.969810	
9	0.00550	0.079422	0.699757	-0.002360	0.962982	
10	0.00925	0.096104	0.846730	-0.001674	0.980754	
11	0.01075	0.094714	0.586462	-0.007580	0.926226	
12	0.00950	0.101333	0.892805	-0.001141	0.986461	

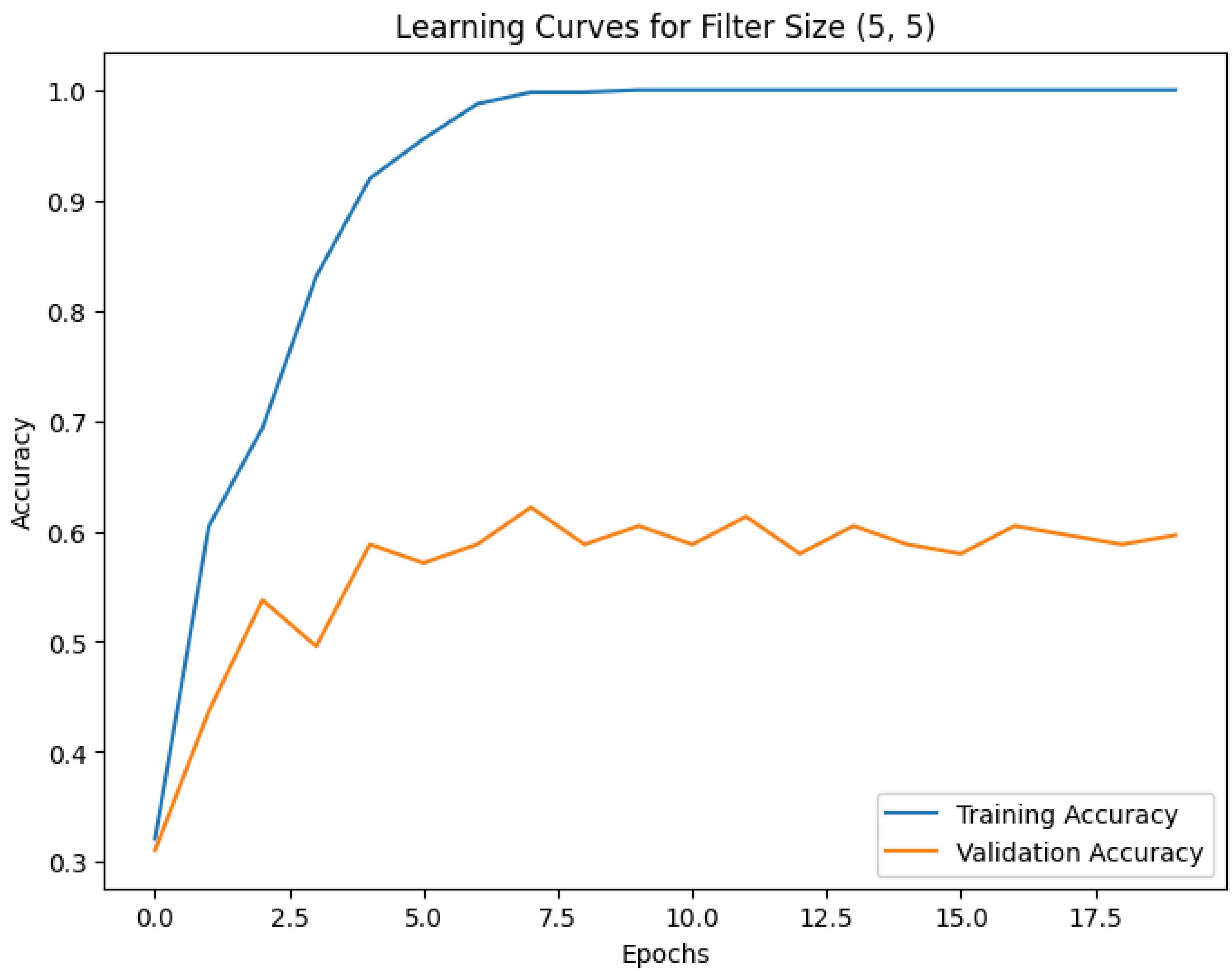
13	0.00750	0.140845	0.872106	-0.001100	0.975959
14	0.00575	0.126374	0.782499	-0.001598	0.959792
15	0.00500	0.149254	0.924172	-0.000410	0.985605
16	0.00625	0.090253	0.795178	-0.001610	0.974446
17	0.00600	0.100840	0.888461	-0.000753	0.985921
18	0.00800	0.083117	0.732307	-0.002924	0.966863
19	0.00625	0.091241	0.803884	-0.001525	0.975506
20	0.01475	0.091331	0.804681	-0.003580	0.975603
21	0.01475	0.129956	0.804681	-0.003580	0.963744
22	0.00950	0.101333	0.892805	-0.001141	0.986461
23	0.00700	0.101083	0.625901	-0.004184	0.932789
24	0.00800	0.134454	0.832531	-0.001609	0.968752
25	0.00575	0.125683	0.778223	-0.001639	0.959034
26	0.01025	0.106494	0.659403	-0.005294	0.938438
27	0.00650	0.094891	0.587557	-0.004563	0.926407
28	0.00575	0.131429	0.813799	-0.001316	0.965378
29	0.01250	0.133333	0.825593	-0.002641	0.967500

	zhangs_metric_y
0	0.186709
1	0.059752
2	-0.096499
3	-0.351258
4	-0.218779
5	-0.131587
6	0.103063
7	-0.265994
8	-0.265994
9	-0.315533
10	-0.166869
11	-0.443027
12	-0.116987
13	-0.134123
14	-0.225530
15	-0.078251
16	-0.216758
17	-0.117764
18	-0.287992
19	-0.207544
20	-0.224493
21	-0.214951

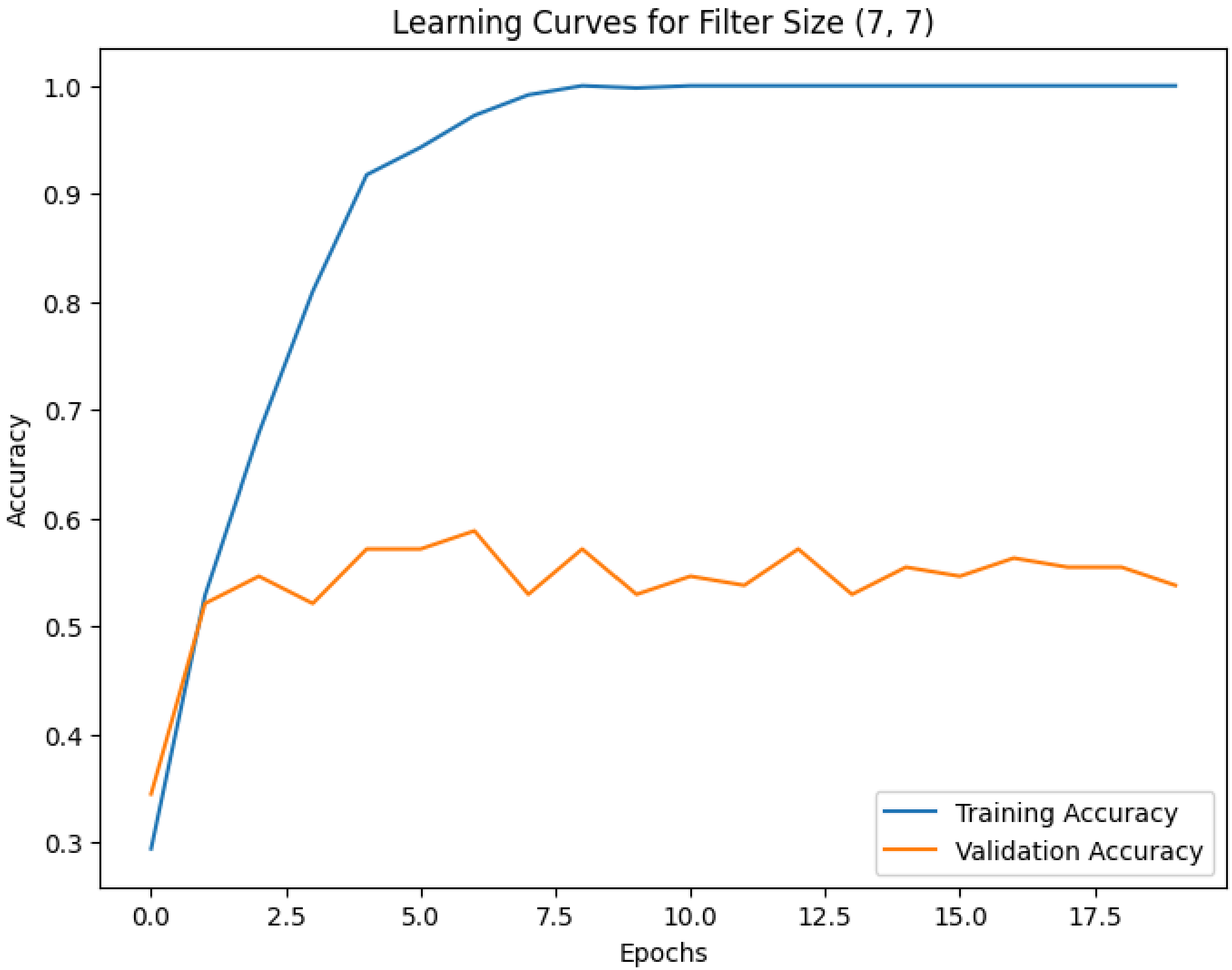
```
22        -0.116987
23        -0.391048
24        -0.176197
25        -0.229964
26        -0.363679
27        -0.429739
28        -0.193075
29        -0.189038
Loading Data:   0%|          | 0/4 [00:00<?, ?it/s]
(128, 128, 3)
Training model with filter size (3, 3)...
Training Progress:   0%|          | 0/20 [00:00<?, ?it/s]
```




```
Performance for Filter Size (3, 3):
Training Loss: 0.8307, Training Accuracy: 63.68%

Training model with filter size (5, 5)...
Training Progress:   0%|          | 0/20 [00:00<?, ?it/s]
```




```
Performance for Filter Size (5, 5):
Training Loss: 0.2961, Training Accuracy: 91.89%

Training model with filter size (7, 7)...
Training Progress:   0%|          | 0/20 [00:00<?, ?it/s]
```




```
Performance for Filter Size (7, 7):
Training Loss: 0.4172, Training Accuracy: 90.71%
```

In [63]:
```
# Based on the given performance metrics and learning curves, we can make several observations:
# The model with the (3, 3) filter size has the highest training accuracy of 63.68% and a
# training loss of 0.2961. A training loss of 0.2723 and a training accuracy of 91.89% are displayed by the model with the (5,
# The model with the (7, 7) filter size records a training accuracy of 90.71% and a training loss
```

```
# of 0.4172.
# The comparison between models trained with different filter sizes (3x3, 5x5, and 7x7) reveals insights into their performanc
# Analyzing learning curves and evaluation metrics indicates whether each model is overfitting, underfitting, or performing op
# Overfitting is signaled by a large gap between training and validation accuracy, with lower validation accuracy and higher l
# Underfitting is indicated by low accuracies and high loss for both training and validation.
# Conversely, models exhibiting similar high accuracies and low losses for both training and validation are deemed well-balanc
# Adjustments to architecture or hyperparameters may be made based on these observations to improve model performance.
```

In [ ]:

```

```