מטלה 2

מגיש: אור שני 206812034

ניסוי 1:

תוצאה:

Epoch 46/50

1875/1875 ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 6s 3ms/step - loss: 0.1158 - sparse_categorical_accuracy: 0.9722 - val_loss: 0.1196 - val_sparse_categorical_accuracy: 0.9736

שכבות:

```
layers = [
  tf.keras.layers.Flatten(input_shape=image_shape),

  tf.keras.layers.Dense(neurons_layer_1,kernel_regularizer=tf.keras.regularizers.l2(0.00001)),
  tf.keras.layers.BatchNormalization(),
  tf.keras.layers.Activation('relu'),

  tf.keras.layers.Dense(neurons_layer_2,kernel_regularizer=tf.keras.regularizers.l2(0.0001)),
  tf.keras.layers.BatchNormalization(),
  tf.keras.layers.Activation('relu'),

  tf.keras.layers.Dense(neurons_layer_3,kernel_regularizer=tf.keras.regularizers.l2(0.001)),
  tf.keras.layers.BatchNormalization(),
  tf.keras.layers.Activation('relu'),

  tf.keras.layers.Dense(neurons_layer_4,kernel_regularizer=tf.keras.regularizers.l2(0.01)),
  tf.keras.layers.BatchNormalization(),
  tf.keras.layers.Activation('relu'),

  tf.keras.layers.Dense(num_of_classes),
  tf.keras.layers.Softmax()
  ]
```

ניסוי 2:

תוצאה:

Epoch 37/50

1875/1875 ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 11s 4ms/step - loss: 0.1616 - sparse_categorical_accuracy: 0.9678 - val_loss: 0.1541 - val_sparse_categorical_accuracy: 0.9739

שכבות:

```
layers = [
  tf.keras.layers.Flatten(input_shape=image_shape),

  tf.keras.layers.Dense(neurons_layer_1,kernel_regularizer=tf.keras.regularizers.l1(0.00001)),
  tf.keras.layers.BatchNormalization(),
  tf.keras.layers.Activation('relu'),

  tf.keras.layers.Dense(neurons_layer_2,kernel_regularizer=tf.keras.regularizers.l1(0.0001)),
  tf.keras.layers.BatchNormalization(),
  tf.keras.layers.Activation('relu'),

  tf.keras.layers.Dense(neurons_layer_3,kernel_regularizer=tf.keras.regularizers.l1(0.001)),
  tf.keras.layers.BatchNormalization(),
  tf.keras.layers.Activation('relu'),

  tf.keras.layers.Dense(neurons_layer_4,kernel_regularizer=tf.keras.regularizers.l1(0.01)),
  tf.keras.layers.BatchNormalization(),
  tf.keras.layers.Activation('relu'),

  tf.keras.layers.Dense(num_of_classes),
  tf.keras.layers.Softmax()
  ]
```

ניסוי 3:

תוצאה:

Epoch 45/50

1875/1875 ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 8s 4ms/step - loss: 0.1255 - sparse_categorical_accuracy: 0.9695 - val_loss: 0.1286 - val_sparse_categorical_accuracy: 0.9713

שכבות:

```
layers = [
  tf.keras.layers.Flatten(input_shape=image_shape),

  tf.keras.layers.Dense(neurons_layer_1,kernel_regularizer=tf.keras.regularizers.l2(0.00001)),
  tf.keras.layers.BatchNormalization(),
  tf.keras.layers.Activation('sigmoid'),

  tf.keras.layers.Dense(neurons_layer_2,kernel_regularizer=tf.keras.regularizers.l2(0.0001)),
  tf.keras.layers.BatchNormalization(),
  tf.keras.layers.Activation('sigmoid'),

  tf.keras.layers.Dense(neurons_layer_3,kernel_regularizer=tf.keras.regularizers.l2(0.001)),
  tf.keras.layers.BatchNormalization(),
  tf.keras.layers.Activation('sigmoid'),

  tf.keras.layers.Dense(neurons_layer_4,kernel_regularizer=tf.keras.regularizers.l2(0.01)),
  tf.keras.layers.BatchNormalization(),
  tf.keras.layers.Activation('sigmoid'),

  tf.keras.layers.Dense(num_of_classes),
  tf.keras.layers.Softmax()
  ]
```

ניסוי 4:

תוצאה:

Epoch 50/50

1875/1875 ━━ 9s 5ms/step - loss: 0.4999 - sparse_categorical_accuracy: 0.8725 - val_loss: 0.1852 - val_sparse_categorical_accuracy: 0.9601

שכבות:

```
layers = [
  tf.keras.layers.Flatten(input_shape=image_shape),

  tf.keras.layers.Dense(neurons_layer_1,kernel_regularizer=tf.keras.regularizers.l2(0.00001)),
  tf.keras.layers.BatchNormalization(),
  tf.keras.layers.Activation('sigmoid'),
  tf.keras.layers.Dropout(0.1),

  tf.keras.layers.Dense(neurons_layer_2,kernel_regularizer=tf.keras.regularizers.l2(0.0001)),
  tf.keras.layers.BatchNormalization(),
  tf.keras.layers.Activation('sigmoid'),
  tf.keras.layers.Dropout(0.1),

  tf.keras.layers.Dense(neurons_layer_3,kernel_regularizer=tf.keras.regularizers.l2(0.001)),
  tf.keras.layers.BatchNormalization(),
  tf.keras.layers.Activation('sigmoid'),
  tf.keras.layers.Dropout(0.1),

  tf.keras.layers.Dense(neurons_layer_4,kernel_regularizer=tf.keras.regularizers.l2(0.01)),
  tf.keras.layers.BatchNormalization(),
  tf.keras.layers.Activation('sigmoid'),
  tf.keras.layers.Dropout(0.1),

  tf.keras.layers.Dense(num_of_classes),
  tf.keras.layers.Softmax()
  ]
```

ניסוי 5:

תוצאה:

Epoch 31/50

1875/1875 ━━ 5s 2ms/step - loss: 0.0529 - sparse_categorical_accuracy: 0.9867 - val_loss: 0.2290 - val_sparse_categorical_accuracy: 0.9680

שכבות:

```
layers = [
  tf.keras.layers.Flatten(input_shape=image_shape),

  tf.keras.layers.Dense(neurons_layer_1),
  tf.keras.layers.Activation('relu'),

  tf.keras.layers.Dense(neurons_layer_2),
  tf.keras.layers.Activation('relu'),

  tf.keras.layers.Dense(neurons_layer_3),
  tf.keras.layers.Activation('relu'),

  tf.keras.layers.Dense(neurons_layer_4),
  tf.keras.layers.Activation('relu'),

  tf.keras.layers.Dense(num_of_classes),
  tf.keras.layers.Softmax()
  ]
```

ניסוי 6:

תוצאה:

Epoch 45/50

1875/1875 ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 6s 3ms/step - loss: 0.0811 - sparse_categorical_accuracy: 0.9741 - val_loss: 0.1034 - val_sparse_categorical_accuracy: 0.9726

שכבות:

```
layers = [
  tf.keras.layers.Flatten(input_shape=image_shape),

  tf.keras.layers.Dense(neurons_layer_1),
  tf.keras.layers.BatchNormalization(),
  tf.keras.layers.Activation('sigmoid'),

  tf.keras.layers.Dense(neurons_layer_2),
  tf.keras.layers.BatchNormalization(),
  tf.keras.layers.Activation('sigmoid'),

  tf.keras.layers.Dense(neurons_layer_3),
  tf.keras.layers.BatchNormalization(),
  tf.keras.layers.Activation('sigmoid'),

  tf.keras.layers.Dense(neurons_layer_4),
  tf.keras.layers.BatchNormalization(),
  tf.keras.layers.Activation('sigmoid'),

  tf.keras.layers.Dense(num_of_classes),
  tf.keras.layers.Softmax()
  ]
```

ניסוי 7:

תוצאה:

Epoch 47/50

1875/1875 ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 6s 3ms/step - loss: 0.0722 - sparse_categorical_accuracy: 0.9780 - val_loss: 0.0998 - val_sparse_categorical_accuracy: 0.9739

שכבות:

```
layers = [
  tf.keras.layers.Flatten(input_shape=image_shape),

  tf.keras.layers.Dense(neurons_layer_1),
  tf.keras.layers.BatchNormalization(),
  tf.keras.layers.Activation('tanh'),

  tf.keras.layers.Dense(neurons_layer_2),
  tf.keras.layers.BatchNormalization(),
  tf.keras.layers.Activation('tanh'),

  tf.keras.layers.Dense(neurons_layer_3),
  tf.keras.layers.BatchNormalization(),
  tf.keras.layers.Activation('tanh'),

  tf.keras.layers.Dense(neurons_layer_4),
  tf.keras.layers.BatchNormalization(),
  tf.keras.layers.Activation('tanh'),

  tf.keras.layers.Dense(num_of_classes),
  tf.keras.layers.Softmax()
  ]
```

ניסוי 8:

תוצאה:

Epoch 43/50

1875/1875 ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 6s 3ms/step - loss: 0.1763 - sparse_categorical_accuracy: 0.9657 - val_loss: 0.1627 - val_sparse_categorical_accuracy: 0.9710

שכבות:

```
layers = [
  tf.keras.layers.Flatten(input_shape=image_shape),

  tf.keras.layers.Dense(neurons_layer_1,kernel_regularizer=tf.keras.regularizers.l1(0.00001)),
  tf.keras.layers.BatchNormalization(),
  tf.keras.layers.Activation('tanh'),

  tf.keras.layers.Dense(neurons_layer_2,kernel_regularizer=tf.keras.regularizers.l1(0.0001)),
  tf.keras.layers.BatchNormalization(),
  tf.keras.layers.Activation('tanh'),

  tf.keras.layers.Dense(neurons_layer_3,kernel_regularizer=tf.keras.regularizers.l1(0.001)),
  tf.keras.layers.BatchNormalization(),
  tf.keras.layers.Activation('tanh'),

  tf.keras.layers.Dense(neurons_layer_4,kernel_regularizer=tf.keras.regularizers.l1(0.01)),
  tf.keras.layers.BatchNormalization(),
  tf.keras.layers.Activation('tanh'),

  tf.keras.layers.Dense(num_of_classes),
  tf.keras.layers.Softmax()
  ]
```

ניסוי 9:

תוצאה:

Epoch 38/50

1875/1875 ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 6s 3ms/step - loss: 0.1389 - sparse_categorical_accuracy: 0.9661 - val_loss: 0.1355 - val_sparse_categorical_accuracy: 0.9720

שכבות:

```
layers = [
  tf.keras.layers.Flatten(input_shape=image_shape),

  tf.keras.layers.Dense(neurons_layer_1,kernel_regularizer=tf.keras.regularizers.l2(0.00001)),
  tf.keras.layers.BatchNormalization(),
  tf.keras.layers.Activation('tanh'),

  tf.keras.layers.Dense(neurons_layer_2,kernel_regularizer=tf.keras.regularizers.l2(0.0001)),
  tf.keras.layers.BatchNormalization(),
  tf.keras.layers.Activation('tanh'),

  tf.keras.layers.Dense(neurons_layer_3,kernel_regularizer=tf.keras.regularizers.l2(0.001)),
  tf.keras.layers.BatchNormalization(),
  tf.keras.layers.Activation('tanh'),

  tf.keras.layers.Dense(neurons_layer_4,kernel_regularizer=tf.keras.regularizers.l2(0.01)),
  tf.keras.layers.BatchNormalization(),
  tf.keras.layers.Activation('tanh'),

  tf.keras.layers.Dense(num_of_classes),
  tf.keras.layers.Softmax()
  ]
```

ניסוי 10:

תוצאה:

Epoch 19/50

1875/1875 ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 4s 2ms/step - loss: 0.0589 - sparse_categorical_accuracy: 0.9828 - val_loss: 0.1387 - val_sparse_categorical_accuracy: 0.9672

שכבות:

```
layers = [
  tf.keras.layers.Flatten(input_shape=image_shape),

  tf.keras.layers.Dense(neurons_layer_1),
  tf.keras.layers.Activation('tanh'),

  tf.keras.layers.Dense(neurons_layer_2),
  tf.keras.layers.Activation('tanh'),

  tf.keras.layers.Dense(neurons_layer_3),
  tf.keras.layers.Activation('tanh'),

  tf.keras.layers.Dense(neurons_layer_4),
  tf.keras.layers.Activation('tanh'),

  tf.keras.layers.Dense(num_of_classes),
  tf.keras.layers.Softmax()
  ]
```

ניסוי 11:

תוצאה:

Epoch 46/50

1875/1875 ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 6s 3ms/step - loss: 0.1352 - sparse_categorical_accuracy: 0.9628 - val_loss: 0.1165 - val_sparse_categorical_accuracy: 0.9703

שכבות:

```
layers = [
  tf.keras.layers.Flatten(input_shape=image_shape),

  tf.keras.layers.Dense(neurons_layer_1),
  tf.keras.layers.Activation('tanh'),
  tf.keras.layers.Dropout(0.1),

  tf.keras.layers.Dense(neurons_layer_2),
  tf.keras.layers.BatchNormalization(),
  tf.keras.layers.Activation('tanh'),

  tf.keras.layers.Dense(neurons_layer_3),
  tf.keras.layers.BatchNormalization(),
  tf.keras.layers.Activation('tanh'),

  tf.keras.layers.Dense(neurons_layer_4),
  tf.keras.layers.Activation('tanh'),
  tf.keras.layers.Dropout(0.1),

  tf.keras.layers.Dense(num_of_classes),
  tf.keras.layers.Softmax()
  ]
```

ניסוי 12:

תוצאה:

Epoch 37/50

1875/1875 ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 4s 2ms/step - loss: 0.0803 - sparse_categorical_accuracy: 0.9756 - val_loss: 0.0954 - val_sparse_categorical_accuracy: 0.9749

שכבות:

```
layers = [
  tf.keras.layers.Flatten(input_shape=image_shape),

  tf.keras.layers.Dense(neurons_layer_1),
  tf.keras.layers.BatchNormalization(),
  tf.keras.layers.Activation('relu'),

  tf.keras.layers.Dense(neurons_layer_2),
  tf.keras.layers.Activation('tanh'),

  tf.keras.layers.Dense(neurons_layer_3),
  tf.keras.layers.BatchNormalization(),
  tf.keras.layers.Activation('sigmoid'),

  tf.keras.layers.Dense(neurons_layer_4),
  tf.keras.layers.Activation('tanh'),

  tf.keras.layers.Dense(num_of_classes),
  tf.keras.layers.Softmax()
  ]
```

ניסוי 13:

תוצאה:

Epoch 49/50

1875/1875 ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 10s 3ms/step - loss: 0.0712 - sparse_categorical_accuracy: 0.9775 - val_loss: 0.0985 - val_sparse_categorical_accuracy: 0.9745

שכבות:

```
layers = [
  tf.keras.layers.Flatten(input_shape=image_shape),

  tf.keras.layers.Dense(neurons_layer_1),
  tf.keras.layers.BatchNormalization(),
  tf.keras.layers.Activation('tanh'),

  tf.keras.layers.Dense(neurons_layer_2),
  tf.keras.layers.BatchNormalization(),
  tf.keras.layers.Activation('sigmoid'),

  tf.keras.layers.Dense(neurons_layer_3),
  tf.keras.layers.BatchNormalization(),
  tf.keras.layers.Activation('sigmoid'),

  tf.keras.layers.Dense(neurons_layer_4),
  tf.keras.layers.BatchNormalization(),
  tf.keras.layers.Activation('tanh'),

  tf.keras.layers.Dense(num_of_classes),
  tf.keras.layers.Softmax()
  ]
```

ניסוי 14:

תוצאה:

Epoch 36/50

1875/1875 ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 9s 2ms/step - loss: 0.0690 - sparse_categorical_accuracy: 0.9802 - val_loss: 0.1367 - val_sparse_categorical_accuracy: 0.9701

שכבות:

```
layers = [
  tf.keras.layers.Flatten(input_shape=image_shape),

  tf.keras.layers.Dense(neurons_layer_1),
  tf.keras.layers.Activation('tanh'),

  tf.keras.layers.Dense(neurons_layer_2),
  tf.keras.layers.BatchNormalization(),
  tf.keras.layers.Activation('relu'),

  tf.keras.layers.Dense(neurons_layer_3),
  tf.keras.layers.BatchNormalization(),
  tf.keras.layers.Activation('relu'),

  tf.keras.layers.Dense(neurons_layer_4),
  tf.keras.layers.Activation('tanh'),

  tf.keras.layers.Dense(num_of_classes),
  tf.keras.layers.Softmax()
  ]
```

ניסוי 15:

תוצאה:

Epoch 20/50

1875/1875 ━━ 6s 3ms/step - loss: 0.0845 - sparse_categorical_accuracy: 0.9765 - val_loss: 0.1167 - val_sparse_categorical_accuracy: 0.9716

שכבות:

```
layers = [
  tf.keras.layers.Flatten(input_shape=image_shape),

  tf.keras.layers.Dense(neurons_layer_1),
  tf.keras.layers.Activation('sigmoid'),

  tf.keras.layers.Dense(neurons_layer_2),
  tf.keras.layers.BatchNormalization(),
  tf.keras.layers.Activation('tanh'),

  tf.keras.layers.Dense(neurons_layer_3),
  tf.keras.layers.BatchNormalization(),
  tf.keras.layers.Activation('relu'),

  tf.keras.layers.Dense(neurons_layer_4),
  tf.keras.layers.Activation('sigmoid'),

  tf.keras.layers.Dense(num_of_classes),
  tf.keras.layers.Softmax()
  ]
```