In [1]:

```python
# rmsprop_optimizer.py

import tensorflow as tf
import numpy as np
import matplotlib.pyplot as plt
from tensorflow.keras.datasets import cifar10
from tensorflow.keras import layers, models
from tensorflow.keras.optimizers import RMSprop
from sklearn.metrics import confusion_matrix
import seaborn as sns
import time
from sklearn.model_selection import train_test_split
```

In [2]:

```python
# Load CIFAR-10 dataset
(X_train, y_train), (X_test, y_test) = cifar10.load_data()

# Normalize the images
X_train = X_train.astype('float32') / 255.0
X_test = X_test.astype('float32') / 255.0

# Ensure labels are integers (no one-hot encoding)
y_train = y_train.astype('int')
y_test = y_test.astype('int')

X_train, X_val, y_train, y_val = train_test_split(X_train, y_train, test_size=0.2,
print (X_train.shape )
print (X_test.shape )
```

```
(40000, 32, 32, 3)
(10000, 32, 32, 3)
```

In [3]:

```python
# Create model function
def create_model(optimizer):
    model = models.Sequential()
    model.add(layers.Flatten(input_shape=(32, 32, 3)))
    model.add(layers.Dense(128, activation='relu'))
    model.add(layers.Dense(10, activation='softmax'))
    model.compile(optimizer=optimizer, loss='sparse_categorical_crossentropy', metr
    return model

# RMSProp optimizer
optimizer = RMSprop(learning_rate=0.001)
```

In [4]:

```python
# Train and evaluate model
start_time = time.time()
model = create_model(optimizer)
history = model.fit(X_train, y_train, epochs=50, batch_size=64, validation_data=(X_
end_time = time.time()

# Record training time
training_time = end_time - start_time
print(f"Training time: {training_time:.2f} seconds")
```

```
c:\Users\Omar Wessam\AppData\Local\Programs\Python\Python312\Lib\site-packages\keras
\src\layers\reshaping\flatten.py:37: UserWarning: Do not pass an `input_shape`/`inpu
t_dim` argument to a layer. When using Sequential models, prefer using an `Input(sha
pe)` object as the first layer in the model instead.
  super().__init__(**kwargs)
```

```
Epoch 1/50
625/625 ━━━━━━━━━━━━━━━━━━━━ 3s 4ms/step - accuracy: 0.2247 - loss: 2.3583 - val_acc
uracy: 0.3204 - val_loss: 1.8890
Epoch 2/50
625/625 ━━━━━━━━━━━━━━━━━━━━ 2s 4ms/step - accuracy: 0.3481 - loss: 1.8342 - val_acc
uracy: 0.3772 - val_loss: 1.7466
Epoch 3/50
625/625 ━━━━━━━━━━━━━━━━━━━━ 2s 3ms/step - accuracy: 0.3733 - loss: 1.7579 - val_acc
uracy: 0.3532 - val_loss: 1.8159
Epoch 4/50
625/625 ━━━━━━━━━━━━━━━━━━━━ 2s 3ms/step - accuracy: 0.3824 - loss: 1.7227 - val_acc
uracy: 0.4046 - val_loss: 1.6998
Epoch 5/50
625/625 ━━━━━━━━━━━━━━━━━━━━ 2s 3ms/step - accuracy: 0.4013 - loss: 1.6873 - val_acc
uracy: 0.4001 - val_loss: 1.6834
Epoch 6/50
625/625 ━━━━━━━━━━━━━━━━━━━━ 2s 3ms/step - accuracy: 0.4110 - loss: 1.6658 - val_acc
uracy: 0.3810 - val_loss: 1.7284
Epoch 7/50
625/625 ━━━━━━━━━━━━━━━━━━━━ 2s 3ms/step - accuracy: 0.4184 - loss: 1.6378 - val_acc
uracy: 0.4124 - val_loss: 1.6657
Epoch 8/50
625/625 ━━━━━━━━━━━━━━━━━━━━ 2s 3ms/step - accuracy: 0.4295 - loss: 1.6220 - val_acc
uracy: 0.4004 - val_loss: 1.7128
Epoch 9/50
625/625 ━━━━━━━━━━━━━━━━━━━━ 2s 3ms/step - accuracy: 0.4260 - loss: 1.6198 - val_acc
uracy: 0.4088 - val_loss: 1.6566
Epoch 10/50
625/625 ━━━━━━━━━━━━━━━━━━━━ 2s 3ms/step - accuracy: 0.4353 - loss: 1.5982 - val_acc
uracy: 0.4382 - val_loss: 1.5962
Epoch 11/50
625/625 ━━━━━━━━━━━━━━━━━━━━ 2s 3ms/step - accuracy: 0.4411 - loss: 1.5741 - val_acc
uracy: 0.4210 - val_loss: 1.6754
Epoch 12/50
625/625 ━━━━━━━━━━━━━━━━━━━━ 2s 3ms/step - accuracy: 0.4408 - loss: 1.5802 - val_acc
uracy: 0.4299 - val_loss: 1.6011
Epoch 13/50
625/625 ━━━━━━━━━━━━━━━━━━━━ 2s 3ms/step - accuracy: 0.4462 - loss: 1.5648 - val_acc
uracy: 0.4326 - val_loss: 1.6131
Epoch 14/50
625/625 ━━━━━━━━━━━━━━━━━━━━ 2s 3ms/step - accuracy: 0.4489 - loss: 1.5611 - val_acc
uracy: 0.4268 - val_loss: 1.6481
Epoch 15/50
625/625 ━━━━━━━━━━━━━━━━━━━━ 2s 3ms/step - accuracy: 0.4520 - loss: 1.5434 - val_acc
uracy: 0.4367 - val_loss: 1.6035
Epoch 16/50
625/625 ━━━━━━━━━━━━━━━━━━━━ 2s 3ms/step - accuracy: 0.4594 - loss: 1.5259 - val_acc
uracy: 0.4451 - val_loss: 1.5793
Epoch 17/50
625/625 ━━━━━━━━━━━━━━━━━━━━ 2s 3ms/step - accuracy: 0.4645 - loss: 1.5189 - val_acc
uracy: 0.4430 - val_loss: 1.5951
Epoch 18/50
625/625 ━━━━━━━━━━━━━━━━━━━━ 2s 3ms/step - accuracy: 0.4624 - loss: 1.5219 - val_acc
uracy: 0.4448 - val_loss: 1.5876
Epoch 19/50
625/625 ━━━━━━━━━━━━━━━━━━━━ 2s 4ms/step - accuracy: 0.4672 - loss: 1.5085 - val_acc
```

```
uracy: 0.4492 - val_loss: 1.5573
Epoch 20/50
625/625 ━━━━━━━━━━━━━━━━━━━━ 2s 3ms/step - accuracy: 0.4701 - loss: 1.5073 - val_acc
uracy: 0.4383 - val_loss: 1.6007
Epoch 21/50
625/625 ━━━━━━━━━━━━━━━━━━━━ 2s 3ms/step - accuracy: 0.4704 - loss: 1.4923 - val_acc
uracy: 0.4329 - val_loss: 1.5923
Epoch 22/50
625/625 ━━━━━━━━━━━━━━━━━━━━ 2s 4ms/step - accuracy: 0.4727 - loss: 1.4925 - val_acc
uracy: 0.4513 - val_loss: 1.5621
Epoch 23/50
625/625 ━━━━━━━━━━━━━━━━━━━━ 2s 3ms/step - accuracy: 0.4742 - loss: 1.4884 - val_acc
uracy: 0.4453 - val_loss: 1.5705
Epoch 24/50
625/625 ━━━━━━━━━━━━━━━━━━━━ 2s 3ms/step - accuracy: 0.4787 - loss: 1.4755 - val_acc
uracy: 0.4539 - val_loss: 1.5464
Epoch 25/50
625/625 ━━━━━━━━━━━━━━━━━━━━ 2s 3ms/step - accuracy: 0.4837 - loss: 1.4656 - val_acc
uracy: 0.4394 - val_loss: 1.5992
Epoch 26/50
625/625 ━━━━━━━━━━━━━━━━━━━━ 2s 3ms/step - accuracy: 0.4784 - loss: 1.4723 - val_acc
uracy: 0.4340 - val_loss: 1.6055
Epoch 27/50
625/625 ━━━━━━━━━━━━━━━━━━━━ 2s 3ms/step - accuracy: 0.4878 - loss: 1.4604 - val_acc
uracy: 0.4215 - val_loss: 1.6681
Epoch 28/50
625/625 ━━━━━━━━━━━━━━━━━━━━ 2s 3ms/step - accuracy: 0.4859 - loss: 1.4656 - val_acc
uracy: 0.4499 - val_loss: 1.5724
Epoch 29/50
625/625 ━━━━━━━━━━━━━━━━━━━━ 2s 3ms/step - accuracy: 0.4909 - loss: 1.4437 - val_acc
uracy: 0.4537 - val_loss: 1.5761
Epoch 30/50
625/625 ━━━━━━━━━━━━━━━━━━━━ 2s 3ms/step - accuracy: 0.4894 - loss: 1.4460 - val_acc
uracy: 0.4193 - val_loss: 1.6502
Epoch 31/50
625/625 ━━━━━━━━━━━━━━━━━━━━ 2s 4ms/step - accuracy: 0.4886 - loss: 1.4424 - val_acc
uracy: 0.4539 - val_loss: 1.5739
Epoch 32/50
625/625 ━━━━━━━━━━━━━━━━━━━━ 2s 3ms/step - accuracy: 0.4918 - loss: 1.4358 - val_acc
uracy: 0.4379 - val_loss: 1.6633
Epoch 33/50
625/625 ━━━━━━━━━━━━━━━━━━━━ 2s 3ms/step - accuracy: 0.4921 - loss: 1.4371 - val_acc
uracy: 0.4360 - val_loss: 1.6032
Epoch 34/50
625/625 ━━━━━━━━━━━━━━━━━━━━ 3s 4ms/step - accuracy: 0.4915 - loss: 1.4336 - val_acc
uracy: 0.4336 - val_loss: 1.6219
Epoch 35/50
625/625 ━━━━━━━━━━━━━━━━━━━━ 3s 4ms/step - accuracy: 0.5002 - loss: 1.4258 - val_acc
uracy: 0.4596 - val_loss: 1.5721
Epoch 36/50
625/625 ━━━━━━━━━━━━━━━━━━━━ 2s 3ms/step - accuracy: 0.4919 - loss: 1.4278 - val_acc
uracy: 0.4422 - val_loss: 1.5952
Epoch 37/50
625/625 ━━━━━━━━━━━━━━━━━━━━ 2s 3ms/step - accuracy: 0.4980 - loss: 1.4204 - val_acc
uracy: 0.4517 - val_loss: 1.6032
Epoch 38/50
```

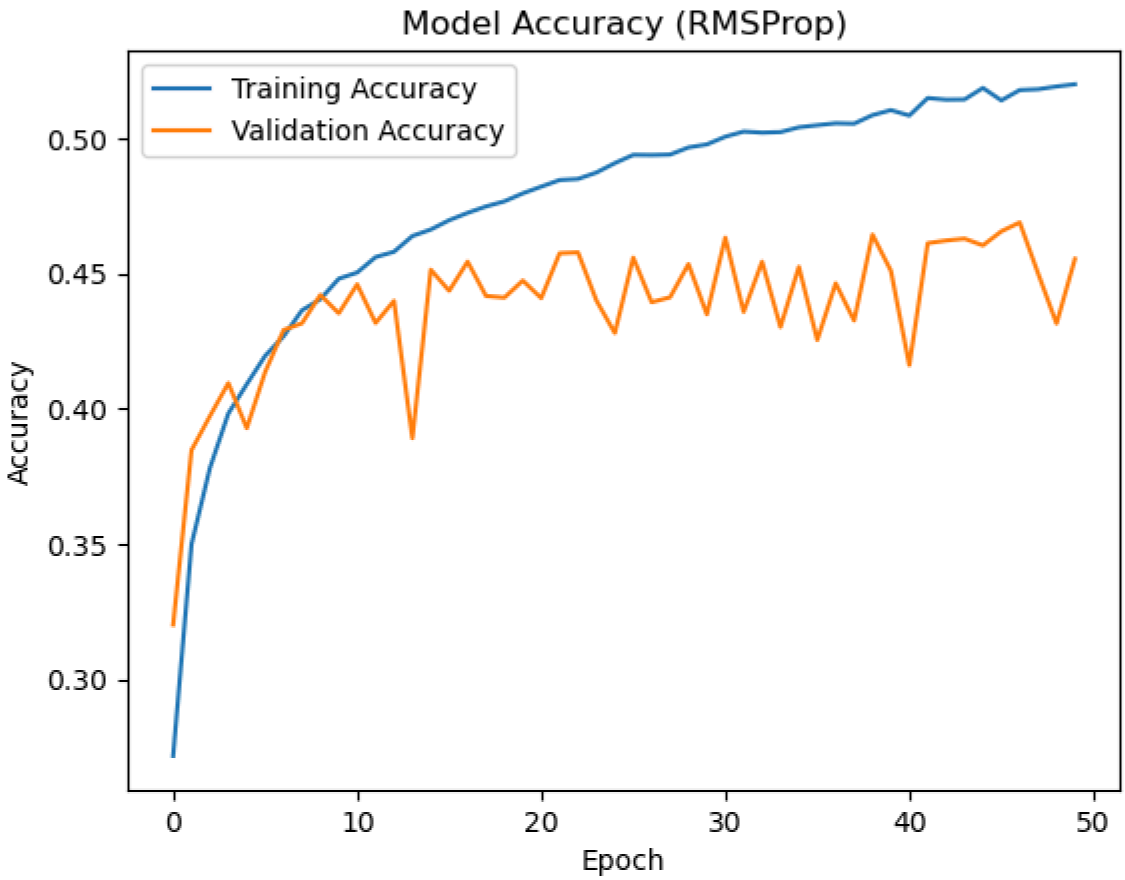
```
625/625 ━━━━━━━━━━━━━━━━━━━━ 2s 3ms/step - accuracy: 0.5042 - loss: 1.4091 - val_acc
uracy: 0.4545 - val_loss: 1.5664
Epoch 39/50
625/625 ━━━━━━━━━━━━━━━━━━━━ 2s 3ms/step - accuracy: 0.5009 - loss: 1.4123 - val_acc
uracy: 0.4466 - val_loss: 1.5927
Epoch 40/50
625/625 ━━━━━━━━━━━━━━━━━━━━ 2s 3ms/step - accuracy: 0.5012 - loss: 1.4139 - val_acc
uracy: 0.4422 - val_loss: 1.5952
Epoch 41/50
625/625 ━━━━━━━━━━━━━━━━━━━━ 2s 3ms/step - accuracy: 0.5022 - loss: 1.4164 - val_acc
uracy: 0.4415 - val_loss: 1.6105
Epoch 42/50
625/625 ━━━━━━━━━━━━━━━━━━━━ 2s 3ms/step - accuracy: 0.5063 - loss: 1.3982 - val_acc
uracy: 0.4435 - val_loss: 1.6421
Epoch 43/50
625/625 ━━━━━━━━━━━━━━━━━━━━ 2s 3ms/step - accuracy: 0.5071 - loss: 1.4123 - val_acc
uracy: 0.4571 - val_loss: 1.5751
Epoch 44/50
625/625 ━━━━━━━━━━━━━━━━━━━━ 2s 3ms/step - accuracy: 0.5086 - loss: 1.4067 - val_acc
uracy: 0.4399 - val_loss: 1.6260
Epoch 45/50
625/625 ━━━━━━━━━━━━━━━━━━━━ 2s 3ms/step - accuracy: 0.5076 - loss: 1.4008 - val_acc
uracy: 0.4393 - val_loss: 1.6957
Epoch 46/50
625/625 ━━━━━━━━━━━━━━━━━━━━ 2s 3ms/step - accuracy: 0.5099 - loss: 1.3874 - val_acc
uracy: 0.4225 - val_loss: 1.6655
Epoch 47/50
625/625 ━━━━━━━━━━━━━━━━━━━━ 2s 3ms/step - accuracy: 0.5000 - loss: 1.4066 - val_acc
uracy: 0.4369 - val_loss: 1.7025
Epoch 48/50
625/625 ━━━━━━━━━━━━━━━━━━━━ 2s 3ms/step - accuracy: 0.5153 - loss: 1.3846 - val_acc
uracy: 0.4133 - val_loss: 1.7446
Epoch 49/50
625/625 ━━━━━━━━━━━━━━━━━━━━ 2s 3ms/step - accuracy: 0.5105 - loss: 1.3847 - val_acc
uracy: 0.4531 - val_loss: 1.5980
Epoch 50/50
625/625 ━━━━━━━━━━━━━━━━━━━━ 2s 3ms/step - accuracy: 0.5111 - loss: 1.3906 - val_acc
uracy: 0.4509 - val_loss: 1.6180
Training time: 98.67 seconds
```

In [8]:

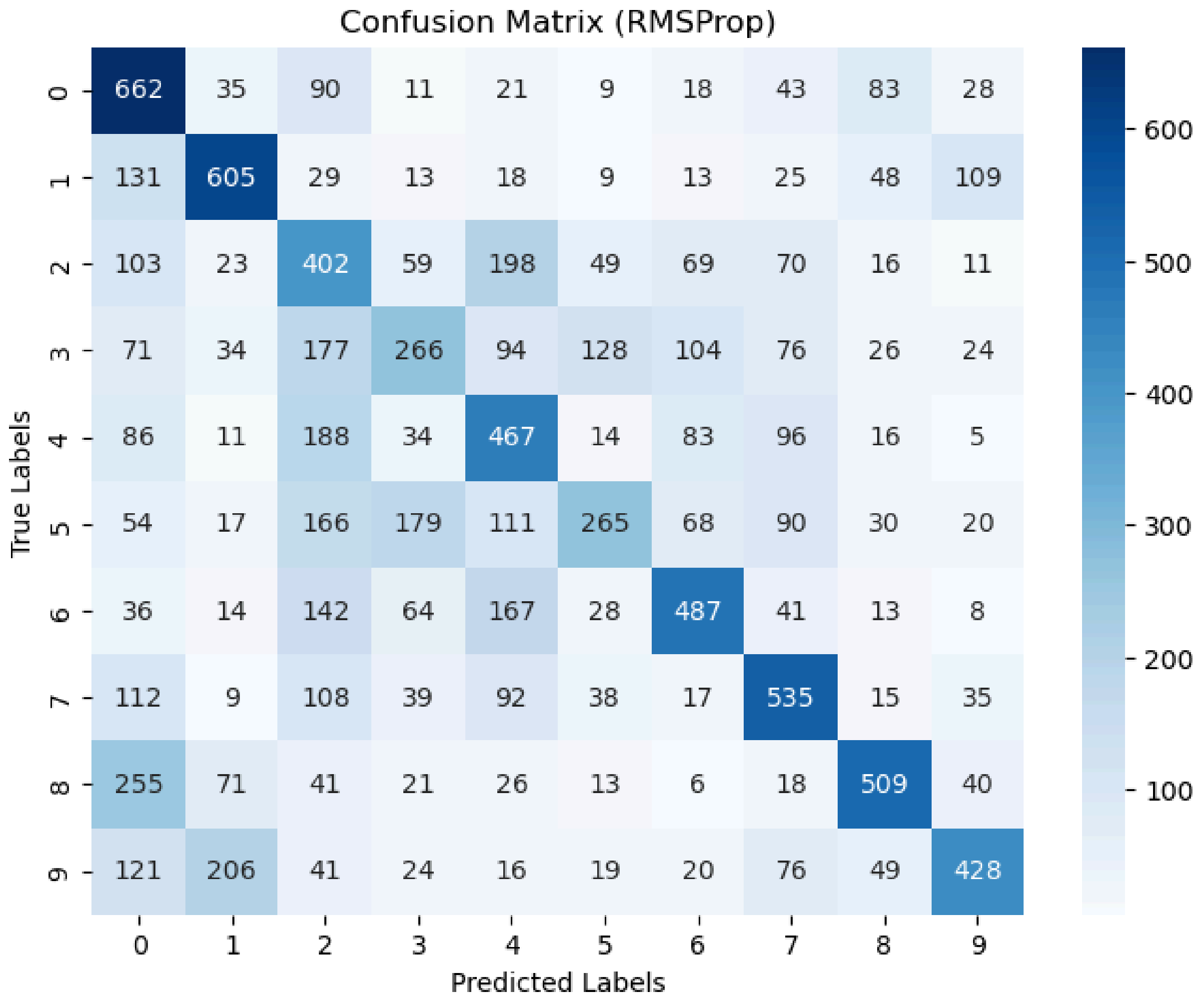
```python
# Plot training and validation accuracy
accuracy = history.history['accuracy']
val_accuracy = history.history['val_accuracy']
plt.plot(accuracy, label='Training Accuracy')
plt.plot(val_accuracy, label='Validation Accuracy')
plt.title('Model Accuracy (RMSProp)')
plt.xlabel('Epoch')
plt.ylabel('Accuracy')
plt.legend()
plt.show()

# Confusion Matrix
y_pred = np.argmax(model.predict(X_test), axis=1)
cm = confusion_matrix(y_test, y_pred)
plt.figure(figsize=(8, 6))
sns.heatmap(cm, annot=True, fmt='d', cmap='Blues', xticklabels=np.arange(10), ytick
```

```
plt.title('Confusion Matrix (RMSProp)')
plt.xlabel('Predicted Labels')
plt.ylabel('True Labels')
plt.show()
```

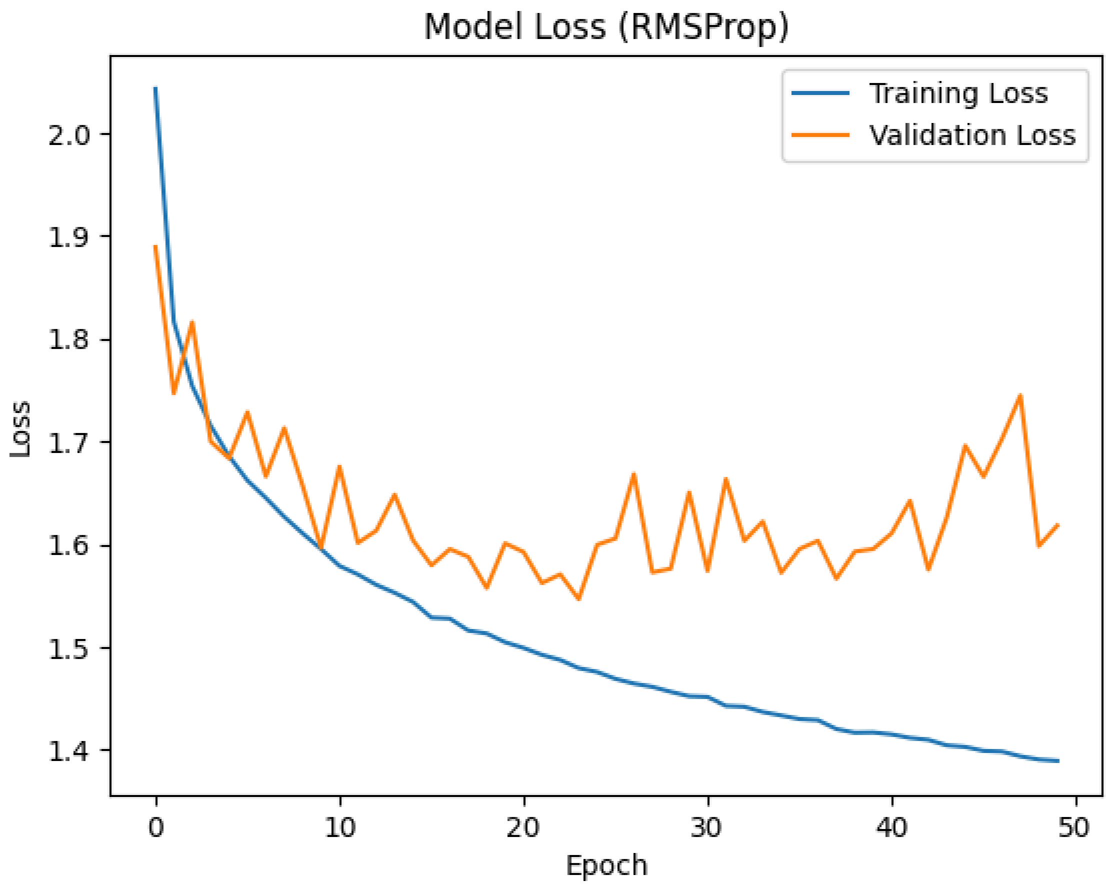



```
313/313 [==============================] - 5s 15ms/step
```



In [5]:

```
# Plot training and validation accuracy
accuracy = history.history['loss']
val_accuracy = history.history['val_loss']
plt.plot(accuracy, label='Training Loss')
plt.plot(val_accuracy, label='Validation Loss')
plt.title('Model Loss (RMSProp)')
plt.xlabel('Epoch')
plt.ylabel('Loss')
plt.legend()
plt.show()
```



Model Loss (RMSProp)
Training Loss
Validation Loss
Loss
Epoch
2.0
1.9
1.8
1.7
1.6
1.5
1.4
0
10
20
30
40
50