In [23]:
```
import tensorflow as tf
from tensorflow.keras import layers, models, optimizers
import matplotlib.pyplot as plt

# Simple, effective data augmentation
def simple_augmentation(image, label):
    image = tf.cast(image, tf.float32)
    # Only horizontal flip - keep it simple
    image = tf.image.random_flip_left_right(image)
    return image, label

# Create dataset pipeline
def create_dataset(images, labels, batch_size, is_training=True, validati
    dataset = tf.data.Dataset.from_tensor_slices((images, labels))

    if is_training:
        # Split training data for validation
        dataset_size = len(images)
        val_size = int(dataset_size * validation_split)
        train_size = dataset_size - val_size

        dataset = dataset.shuffle(10000, seed=42)
        train_dataset = dataset.take(train_size)
        val_dataset = dataset.skip(train_size)

        # Apply minimal augmentation only to training data
        train_dataset = train_dataset.cache()
        train_dataset = train_dataset.shuffle(5000, reshuffle_each_iterat
        train_dataset = train_dataset.map(simple_augmentation, num_parall
        train_dataset = train_dataset.batch(batch_size)
        train_dataset = train_dataset.prefetch(tf.data.AUTOTUNE)

        # Validation dataset without augmentation
        val_dataset = val_dataset.batch(batch_size)
        val_dataset = val_dataset.prefetch(tf.data.AUTOTUNE)

        return train_dataset, val_dataset
    else:
        dataset = dataset.batch(batch_size)
        dataset = dataset.prefetch(tf.data.AUTOTUNE)
        return dataset

# Create a parameter-efficient but effective model
def create_simple_effective_model():
    inputs = layers.Input(shape=(32, 32, 3))

    # Block 1 - smaller filters
    x = layers.Conv2D(24, (3, 3), padding='same', activation='relu')(inpu
    x = layers.BatchNormalization()(x)
    x = layers.Conv2D(24, (3, 3), padding='same', activation='relu')(x)
    x = layers.MaxPooling2D((2, 2))(x)
    x = layers.Dropout(0.25)(x)

    # Block 2 - moderate filters
    x = layers.Conv2D(48, (3, 3), padding='same', activation='relu')(x)
    x = layers.BatchNormalization()(x)
    x = layers.Conv2D(48, (3, 3), padding='same', activation='relu')(x)
    x = layers.MaxPooling2D((2, 2))(x)
```

```
    x = layers.Dropout(0.25)(x)

    # Block 3 - slightly smaller final block
    x = layers.Conv2D(68, (3, 3), padding='same', activation='relu')(x)
    x = layers.BatchNormalization()(x)
    x = layers.Conv2D(68, (3, 3), padding='same', activation='relu')(x)
    x = layers.Dropout(0.25)(x)

    # Global average pooling instead of flatten
    x = layers.GlobalAveragePooling2D()(x)

    # Slightly smaller dense layer
    x = layers.Dense(120, activation='relu')(x)  # 128->120
    x = layers.Dropout(0.5)(x)

    outputs = layers.Dense(10)(x)

    model = models.Model(inputs, outputs, name='competitive_model')
    return model

# Simple cosine decay
def cosine_decay_schedule(epoch, total_epochs=100):
    import math
    return 0.001 * 0.5 * (1 + math.cos(math.pi * epoch / total_epochs))

# Build the model
model = create_simple_effective_model()
model.build(input_shape=(None, 32, 32, 3))

# Compile with basic settings
optimizer = optimizers.Adam(learning_rate=0.001)
model.compile(
    optimizer=optimizer,
    loss=tf.keras.losses.SparseCategoricalCrossentropy(from_logits=True),
    metrics=[
        tf.keras.metrics.SparseCategoricalCrossentropy(from_logits=True,
        tf.keras.metrics.SparseCategoricalAccuracy(name='accuracy')
    ]
)

# Display model summary
model.summary()

# Calculate total parameters
total_params = model.count_params()
print(f"Total Parameters: {total_params:,}")
print(f"Parameter budget used: {total_params/122000*100:.1f}%")

# Simple callbacks focused on generalization
callbacks = [
    tf.keras.callbacks.LearningRateScheduler(cosine_decay_schedule),
    tf.keras.callbacks.EarlyStopping(
        monitor='val_accuracy',  # Monitor accuracy instead
        patience=10,
        restore_best_weights=True,
        mode='max',
        verbose=1
    ),
    tf.keras.callbacks.ModelCheckpoint(
        'best_model.h5',
```

```
        monitor='val_accuracy',
        save_best_only=True,
        mode='max',
        verbose=1
    )
]

# Load and preprocess CIFAR-10 dataset
(train_images, train_labels), (test_images, test_labels) = tf.keras.datas

# Normalize pixel values to be between 0 and 1
train_images = train_images.astype('float32') / 255.0
test_images = test_images.astype('float32') / 255.0

# Create datasets with larger validation split
batch_size = 32  # Smaller batch size
train_dataset, val_dataset = create_dataset(train_images, train_labels, b
test_dataset = create_dataset(test_images, test_labels, batch_size, is_tr

# Train the model with fewer epochs
history = model.fit(
    train_dataset,
    epochs=50,  # Much fewer epochs
    validation_data=val_dataset,
    callbacks=callbacks,
    verbose=1
)

# Evaluate on test set
test_results = model.evaluate(test_dataset, verbose=0)
print(f"\nTest Results:")
print(f"Test Loss: {test_results[0]:.4f}")
print(f"Test Cross-Entropy: {test_results[1]:.4f}")
print(f"Test Accuracy: {test_results[2]:.4f}")

# Plot training history with detailed CE information
plt.figure(figsize=(15, 5))

plt.subplot(1, 3, 1)
plt.plot(history.history['ce'], label='Train CE', alpha=0.8)
plt.plot(history.history['val_ce'], label='Val CE', alpha=0.8)

# Find and annotate lowest validation CE
min_val_ce = min(history.history['val_ce'])
min_val_ce_epoch = history.history['val_ce'].index(min_val_ce)
plt.annotate(f'Lowest Val CE: {min_val_ce:.4f}\nEpoch: {min_val_ce_epoch
             xy=(min_val_ce_epoch, min_val_ce),
             xytext=(min_val_ce_epoch + 5, min_val_ce + 0.1),
             arrowprops=dict(arrowstyle='->', color='red', alpha=0.7),
             bbox=dict(boxstyle="round,pad=0.3", facecolor='yellow', alph
             fontsize=10)

# Mark the point
plt.plot(min_val_ce_epoch, min_val_ce, 'ro', markersize=8, alpha=0.8)

plt.title(f'Cross Entropy (Best Val CE: {min_val_ce:.4f})')
plt.xlabel('Epoch')
plt.ylabel('Cross Entropy')
plt.legend()
plt.grid(True, alpha=0.3)
```

```
plt.subplot(1, 3, 2)
plt.plot(history.history['accuracy'], label='Train Acc', alpha=0.8)
plt.plot(history.history['val_accuracy'], label='Val Acc', alpha=0.8)

# Find and annotate highest validation accuracy
max_val_acc = max(history.history['val_accuracy'])
max_val_acc_epoch = history.history['val_accuracy'].index(max_val_acc)
plt.annotate(f'Best Val Acc: {max_val_acc:.4f}\nEpoch: {max_val_acc_epoch
             xy=(max_val_acc_epoch, max_val_acc),
             xytext=(max_val_acc_epoch + 5, max_val_acc - 0.05),
             arrowprops=dict(arrowstyle='->', color='green', alpha=0.7),
             bbox=dict(boxstyle="round,pad=0.3", facecolor='lightgreen',
             fontsize=10)

# Mark the point
plt.plot(max_val_acc_epoch, max_val_acc, 'go', markersize=8, alpha=0.8)

plt.title(f'Accuracy (Best Val Acc: {max_val_acc:.4f})')
plt.xlabel('Epoch')
plt.ylabel('Accuracy')
plt.legend()
plt.grid(True, alpha=0.3)

plt.subplot(1, 3, 3)
plt.plot(history.history['loss'], label='Train Loss', alpha=0.8)
plt.plot(history.history['val_loss'], label='Val Loss', alpha=0.8)

# Find and annotate lowest validation loss
min_val_loss = min(history.history['val_loss'])
min_val_loss_epoch = history.history['val_loss'].index(min_val_loss)
plt.annotate(f'Lowest Val Loss: {min_val_loss:.4f}\nEpoch: {min_val_loss_
             xy=(min_val_loss_epoch, min_val_loss),
             xytext=(min_val_loss_epoch + 5, min_val_loss + 0.1),
             arrowprops=dict(arrowstyle='->', color='blue', alpha=0.7),
             bbox=dict(boxstyle="round,pad=0.3", facecolor='lightblue', a
             fontsize=10)

# Mark the point
plt.plot(min_val_loss_epoch, min_val_loss, 'bo', markersize=8, alpha=0.8)

plt.title(f'Loss (Best Val Loss: {min_val_loss:.4f})')
plt.xlabel('Epoch')
plt.ylabel('Loss')
plt.legend()
plt.grid(True, alpha=0.3)

plt.tight_layout()
plt.show()

# Print summary of best results
print(f"\n🏆 COMPETITION SUMMARY:")
print(f"📊 Parameters: {total_params:,} / 122,000 ({total_params/122000*1
print(f"🎯 Best Validation CE: {min_val_ce:.4f} (Epoch {min_val_ce_epoch
print(f"🎯 Best Validation Accuracy: {max_val_acc:.4f} (Epoch {max_val_ac
print(f"🎯 Best Validation Loss: {min_val_loss:.4f} (Epoch {min_val_loss_
print(f"📈 Final Test CE: {test_results[1]:.4f}")
print(f"📈 Final Test Accuracy: {test_results[2]:.4f}")
```

```
Model: "competitive_model"
_________________________________________________________________
 Layer (type)                Output Shape              Param #   
=================================================================
 input_23 (InputLayer)       [(None, 32, 32, 3)]       0         
                                                                 
 conv2d_66 (Conv2D)          (None, 32, 32, 24)        672       
                                                                 
 batch_normalization_61 (Bat  (None, 32, 32, 24)       96        
 chNormalization)                                                
                                                                 
 conv2d_67 (Conv2D)          (None, 32, 32, 24)        5208      
                                                                 
 max_pooling2d_28 (MaxPoolin  (None, 16, 16, 24)       0         
 g2D)                                                            
                                                                 
 dropout_62 (Dropout)        (None, 16, 16, 24)        0         
                                                                 
 conv2d_68 (Conv2D)          (None, 16, 16, 48)        10416     
                                                                 
 batch_normalization_62 (Bat  (None, 16, 16, 48)       192       
 chNormalization)                                                
                                                                 
 conv2d_69 (Conv2D)          (None, 16, 16, 48)        20784     
                                                                 
 max_pooling2d_29 (MaxPoolin  (None, 8, 8, 48)         0         
 g2D)                                                            
                                                                 
 dropout_63 (Dropout)        (None, 8, 8, 48)          0         
                                                                 
 conv2d_70 (Conv2D)          (None, 8, 8, 68)          29444     
                                                                 
 batch_normalization_63 (Bat  (None, 8, 8, 68)         272       
 chNormalization)                                                
                                                                 
 conv2d_71 (Conv2D)          (None, 8, 8, 68)          41684     
                                                                 
 dropout_64 (Dropout)        (None, 8, 8, 68)          0         
                                                                 
 global_average_pooling2d_14  (None, 68)               0         
  (GlobalAveragePooling2D)                                       
                                                                 
 dense_30 (Dense)            (None, 120)               8280      
                                                                 
 dropout_65 (Dropout)        (None, 120)               0         
                                                                 
 dense_31 (Dense)            (None, 10)                1210      
                                                                 
=================================================================
Total params: 118,258
Trainable params: 117,978
Non-trainable params: 280
_________________________________________________________________
Total Parameters: 118,258
Parameter budget used: 96.9%
Epoch 1/50
1326/1329 [============================>.] - ETA: 0s - loss: 1.5080 - c
e: 1.5080 - accuracy: 0.4498
Epoch 1: val_accuracy improved from -inf to 0.50853, saving model to bes
t_model.h5
```

```
1329/1329 [==============================] - 10s 7ms/step - loss: 1.5076
- ce: 1.5076 - accuracy: 0.4500 - val_loss: 1.3184 - val_ce: 1.3184 - va
l_accuracy: 0.5085 - lr: 0.0010
Epoch 2/50
1323/1329 [============================>.] - ETA: 0s - loss: 1.1625 - c
e: 1.1625 - accuracy: 0.5934
Epoch 2: val_accuracy improved from 0.50853 to 0.59320, saving model to
best_model.h5
1329/1329 [==============================] - 9s 7ms/step - loss: 1.1620
- ce: 1.1620 - accuracy: 0.5936 - val_loss: 1.1110 - val_ce: 1.1110 - va
l_accuracy: 0.5932 - lr: 0.0010
Epoch 3/50
1326/1329 [============================>.] - ETA: 0s - loss: 1.0422 - c
e: 1.0422 - accuracy: 0.6344
Epoch 3: val_accuracy improved from 0.59320 to 0.66987, saving model to
best_model.h5
1329/1329 [==============================] - 9s 7ms/step - loss: 1.0422
- ce: 1.0422 - accuracy: 0.6343 - val_loss: 0.9074 - val_ce: 0.9074 - va
l_accuracy: 0.6699 - lr: 1.0000e-03
Epoch 4/50
1325/1329 [============================>.] - ETA: 0s - loss: 0.9568 - c
e: 0.9568 - accuracy: 0.6656
Epoch 4: val_accuracy improved from 0.66987 to 0.71413, saving model to
best_model.h5
1329/1329 [==============================] - 9s 7ms/step - loss: 0.9568
- ce: 0.9568 - accuracy: 0.6657 - val_loss: 0.7920 - val_ce: 0.7920 - va
l_accuracy: 0.7141 - lr: 1.0000e-03
Epoch 5/50
1328/1329 [============================>.] - ETA: 0s - loss: 0.8861 - c
e: 0.8861 - accuracy: 0.6895
Epoch 5: val_accuracy improved from 0.71413 to 0.74587, saving model to
best_model.h5
1329/1329 [==============================] - 9s 7ms/step - loss: 0.8861
- ce: 0.8861 - accuracy: 0.6895 - val_loss: 0.7164 - val_ce: 0.7164 - va
l_accuracy: 0.7459 - lr: 1.0000e-03
Epoch 6/50
1327/1329 [============================>.] - ETA: 0s - loss: 0.8384 - c
e: 0.8384 - accuracy: 0.7104
Epoch 6: val_accuracy did not improve from 0.74587
1329/1329 [==============================] - 9s 7ms/step - loss: 0.8383
- ce: 0.8383 - accuracy: 0.7105 - val_loss: 0.7783 - val_ce: 0.7783 - va
l_accuracy: 0.7236 - lr: 1.0000e-03
Epoch 7/50
1321/1329 [============================>.] - ETA: 0s - loss: 0.7997 - c
e: 0.7997 - accuracy: 0.7250
Epoch 7: val_accuracy improved from 0.74587 to 0.76693, saving model to
best_model.h5
1329/1329 [==============================] - 9s 7ms/step - loss: 0.7994
- ce: 0.7994 - accuracy: 0.7252 - val_loss: 0.6629 - val_ce: 0.6629 - va
l_accuracy: 0.7669 - lr: 9.9958e-04
Epoch 8/50
1329/1329 [==============================] - ETA: 0s - loss: 0.7243 - c
e: 0.7243 - accuracy: 0.7509
Epoch 8: val_accuracy improved from 0.76693 to 0.79907, saving model to
best_model.h5
1329/1329 [==============================] - 9s 7ms/step - loss: 0.7243
- ce: 0.7243 - accuracy: 0.7509 - val_loss: 0.5685 - val_ce: 0.5685 - va
l_accuracy: 0.7991 - lr: 1.8408e-05
Epoch 9/50
1323/1329 [============================>.] - ETA: 0s - loss: 0.6957 - c
```

```
e: 0.6957 - accuracy: 0.7601
Epoch 9: val_accuracy improved from 0.79907 to 0.80267, saving model to
best_model.h5
1329/1329 [==============================] - 9s 7ms/step - loss: 0.6954
- ce: 0.6954 - accuracy: 0.7603 - val_loss: 0.5499 - val_ce: 0.5499 - va
l_accuracy: 0.8027 - lr: 7.0725e-05
Epoch 10/50
1328/1329 [============================>.] - ETA: 0s - loss: 0.6813 - c
e: 0.6813 - accuracy: 0.7672
Epoch 10: val_accuracy improved from 0.80267 to 0.81360, saving model to
best_model.h5
1329/1329 [==============================] - 9s 7ms/step - loss: 0.6813
- ce: 0.6813 - accuracy: 0.7672 - val_loss: 0.5288 - val_ce: 0.5288 - va
l_accuracy: 0.8136 - lr: 2.0896e-04
Epoch 11/50
1327/1329 [============================>.] - ETA: 0s - loss: 0.6558 - c
e: 0.6558 - accuracy: 0.7763
Epoch 11: val_accuracy did not improve from 0.81360
1329/1329 [==============================] - 9s 7ms/step - loss: 0.6557
- ce: 0.6557 - accuracy: 0.7764 - val_loss: 0.5278 - val_ce: 0.5278 - va
l_accuracy: 0.8113 - lr: 1.3365e-04
Epoch 12/50
1322/1329 [============================>.] - ETA: 0s - loss: 0.6628 - c
e: 0.6628 - accuracy: 0.7746
Epoch 12: val_accuracy did not improve from 0.81360
1329/1329 [==============================] - 9s 7ms/step - loss: 0.6622
- ce: 0.6622 - accuracy: 0.7748 - val_loss: 0.5331 - val_ce: 0.5331 - va
l_accuracy: 0.8123 - lr: 2.9049e-04
Epoch 13/50
1322/1329 [============================>.] - ETA: 0s - loss: 0.6373 - c
e: 0.6373 - accuracy: 0.7821
Epoch 13: val_accuracy improved from 0.81360 to 0.82307, saving model to
best_model.h5
1329/1329 [==============================] - 9s 7ms/step - loss: 0.6371
- ce: 0.6371 - accuracy: 0.7823 - val_loss: 0.4978 - val_ce: 0.4978 - va
l_accuracy: 0.8231 - lr: 7.9585e-05
Epoch 14/50
1324/1329 [============================>.] - ETA: 0s - loss: 0.7127 - c
e: 0.7127 - accuracy: 0.7555
Epoch 14: val_accuracy did not improve from 0.82307
1329/1329 [==============================] - 9s 7ms/step - loss: 0.7125
- ce: 0.7125 - accuracy: 0.7555 - val_loss: 0.6432 - val_ce: 0.6432 - va
l_accuracy: 0.7677 - lr: 8.7437e-04
Epoch 15/50
1321/1329 [============================>.] - ETA: 0s - loss: 0.6552 - c
e: 0.6552 - accuracy: 0.7771
Epoch 15: val_accuracy improved from 0.82307 to 0.82507, saving model to
best_model.h5
1329/1329 [==============================] - 9s 7ms/step - loss: 0.6547
- ce: 0.6547 - accuracy: 0.7774 - val_loss: 0.5189 - val_ce: 0.5189 - va
l_accuracy: 0.8251 - lr: 5.3975e-04
Epoch 16/50
1328/1329 [============================>.] - ETA: 0s - loss: 0.6568 - c
e: 0.6568 - accuracy: 0.7769
Epoch 16: val_accuracy did not improve from 0.82507
1329/1329 [==============================] - 9s 7ms/step - loss: 0.6568
- ce: 0.6568 - accuracy: 0.7769 - val_loss: 0.5332 - val_ce: 0.5332 - va
l_accuracy: 0.8153 - lr: 6.6456e-04
Epoch 17/50
1320/1329 [============================>.] - ETA: 0s - loss: 0.6799 - c
```

```
e: 0.6799 - accuracy: 0.7688
Epoch 17: val_accuracy did not improve from 0.82507
1329/1329 [==============================] - 9s 7ms/step - loss: 0.6797
- ce: 0.6797 - accuracy: 0.7689 - val_loss: 0.5696 - val_ce: 0.5696 - va
l_accuracy: 0.7987 - lr: 9.9973e-04
Epoch 18/50
1319/1329 [============================>.] - ETA: 0s - loss: 0.6009 - c
e: 0.6009 - accuracy: 0.7948
Epoch 18: val_accuracy improved from 0.82507 to 0.84427, saving model to
best_model.h5
1329/1329 [==============================] - 9s 6ms/step - loss: 0.6009
- ce: 0.6009 - accuracy: 0.7950 - val_loss: 0.4555 - val_ce: 0.4555 - va
l_accuracy: 0.8443 - lr: 2.4941e-04
Epoch 19/50
1322/1329 [============================>.] - ETA: 0s - loss: 0.5770 - c
e: 0.5770 - accuracy: 0.8025
Epoch 19: val_accuracy did not improve from 0.84427
1329/1329 [==============================] - 9s 7ms/step - loss: 0.5768
- ce: 0.5768 - accuracy: 0.8025 - val_loss: 0.4561 - val_ce: 0.4561 - va
l_accuracy: 0.8428 - lr: 2.7019e-04
Epoch 20/50
1327/1329 [============================>.] - ETA: 0s - loss: 0.6152 - c
e: 0.6152 - accuracy: 0.7919
Epoch 20: val_accuracy did not improve from 0.84427
1329/1329 [==============================] - 9s 7ms/step - loss: 0.6152
- ce: 0.6152 - accuracy: 0.7920 - val_loss: 0.4960 - val_ce: 0.4960 - va
l_accuracy: 0.8281 - lr: 6.9197e-04
Epoch 21/50
1328/1329 [============================>.] - ETA: 0s - loss: 0.5780 - c
e: 0.5780 - accuracy: 0.8041
Epoch 21: val_accuracy improved from 0.84427 to 0.84680, saving model to
best_model.h5
1329/1329 [==============================] - 9s 7ms/step - loss: 0.5780
- ce: 0.5780 - accuracy: 0.8041 - val_loss: 0.4429 - val_ce: 0.4429 - va
l_accuracy: 0.8468 - lr: 1.3757e-05
Epoch 22/50
1321/1329 [============================>.] - ETA: 0s - loss: 0.6397 - c
e: 0.6397 - accuracy: 0.7850
Epoch 22: val_accuracy did not improve from 0.84680
1329/1329 [==============================] - 9s 7ms/step - loss: 0.6397
- ce: 0.6397 - accuracy: 0.7847 - val_loss: 0.5232 - val_ce: 0.5232 - va
l_accuracy: 0.8268 - lr: 9.9646e-04
Epoch 23/50
1321/1329 [============================>.] - ETA: 0s - loss: 0.6211 - c
e: 0.6211 - accuracy: 0.7899
Epoch 23: val_accuracy did not improve from 0.84680
1329/1329 [==============================] - 9s 7ms/step - loss: 0.6212
- ce: 0.6212 - accuracy: 0.7899 - val_loss: 0.5717 - val_ce: 0.5717 - va
l_accuracy: 0.7984 - lr: 9.0587e-04
Epoch 24/50
1321/1329 [============================>.] - ETA: 0s - loss: 0.6171 - c
e: 0.6171 - accuracy: 0.7914
Epoch 24: val_accuracy did not improve from 0.84680
1329/1329 [==============================] - 9s 7ms/step - loss: 0.6173
- ce: 0.6173 - accuracy: 0.7914 - val_loss: 0.5023 - val_ce: 0.5023 - va
l_accuracy: 0.8280 - lr: 9.9776e-04
Epoch 25/50
1328/1329 [============================>.] - ETA: 0s - loss: 0.6055 - c
e: 0.6055 - accuracy: 0.7932
Epoch 25: val_accuracy did not improve from 0.84680
```

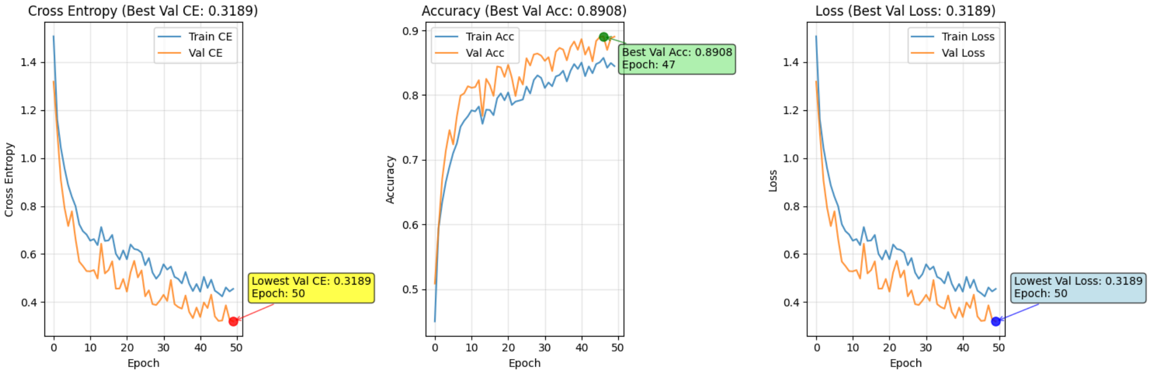
```
1329/1329 [==============================] - 9s 7ms/step - loss: 0.6056
- ce: 0.6056 - accuracy: 0.7932 - val_loss: 0.5316 - val_ce: 0.5316 - va
l_accuracy: 0.8163 - lr: 9.9978e-04
Epoch 26/50
1320/1329 [============================>.] - ETA: 0s - loss: 0.5534 - c
e: 0.5534 - accuracy: 0.8131
Epoch 26: val_accuracy improved from 0.84680 to 0.85693, saving model to
best_model.h5
1329/1329 [==============================] - 9s 7ms/step - loss: 0.5534
- ce: 0.5534 - accuracy: 0.8131 - val_loss: 0.4236 - val_ce: 0.4236 - va
l_accuracy: 0.8569 - lr: 5.3757e-04
Epoch 27/50
1321/1329 [============================>.] - ETA: 0s - loss: 0.5837 - c
e: 0.5837 - accuracy: 0.8025
Epoch 27: val_accuracy did not improve from 0.85693
1329/1329 [==============================] - 9s 7ms/step - loss: 0.5835
- ce: 0.5835 - accuracy: 0.8024 - val_loss: 0.4487 - val_ce: 0.4487 - va
l_accuracy: 0.8457 - lr: 9.5489e-04
Epoch 28/50
1320/1329 [============================>.] - ETA: 0s - loss: 0.5228 - c
e: 0.5228 - accuracy: 0.8232
Epoch 28: val_accuracy improved from 0.85693 to 0.86280, saving model to
best_model.h5
1329/1329 [==============================] - 9s 7ms/step - loss: 0.5226
- ce: 0.5226 - accuracy: 0.8233 - val_loss: 0.3910 - val_ce: 0.3910 - va
l_accuracy: 0.8628 - lr: 3.4819e-04
Epoch 29/50
1324/1329 [============================>.] - ETA: 0s - loss: 0.4970 - c
e: 0.4970 - accuracy: 0.8304
Epoch 29: val_accuracy improved from 0.86280 to 0.86413, saving model to
best_model.h5
1329/1329 [==============================] - 9s 7ms/step - loss: 0.4971
- ce: 0.4971 - accuracy: 0.8304 - val_loss: 0.3876 - val_ce: 0.3876 - va
l_accuracy: 0.8641 - lr: 1.6462e-04
Epoch 30/50
1325/1329 [============================>.] - ETA: 0s - loss: 0.5164 - c
e: 0.5164 - accuracy: 0.8265
Epoch 30: val_accuracy did not improve from 0.86413
1329/1329 [==============================] - 9s 7ms/step - loss: 0.5164
- ce: 0.5164 - accuracy: 0.8265 - val_loss: 0.4059 - val_ce: 0.4059 - va
l_accuracy: 0.8609 - lr: 4.6585e-04
Epoch 31/50
1320/1329 [============================>.] - ETA: 0s - loss: 0.5575 - c
e: 0.5575 - accuracy: 0.8109
Epoch 31: val_accuracy did not improve from 0.86413
1329/1329 [==============================] - 9s 6ms/step - loss: 0.5574
- ce: 0.5574 - accuracy: 0.8110 - val_loss: 0.4300 - val_ce: 0.4300 - va
l_accuracy: 0.8528 - lr: 9.4167e-04
Epoch 32/50
1325/1329 [============================>.] - ETA: 0s - loss: 0.5354 - c
e: 0.5354 - accuracy: 0.8194
Epoch 32: val_accuracy did not improve from 0.86413
1329/1329 [==============================] - 9s 7ms/step - loss: 0.5350
- ce: 0.5350 - accuracy: 0.8195 - val_loss: 0.4044 - val_ce: 0.4044 - va
l_accuracy: 0.8585 - lr: 7.6562e-04
Epoch 33/50
1329/1329 [==============================] - ETA: 0s - loss: 0.5478 - c
e: 0.5478 - accuracy: 0.8137
Epoch 33: val_accuracy did not improve from 0.86413
1329/1329 [==============================] - 9s 7ms/step - loss: 0.5478
```

```
- ce: 0.5478 - accuracy: 0.8137 - val_loss: 0.4918 - val_ce: 0.4918 - va
l_accuracy: 0.8371 - lr: 9.6324e-04
Epoch 34/50
1322/1329 [============================>.] - ETA: 0s - loss: 0.5052 - c
e: 0.5052 - accuracy: 0.8286
Epoch 34: val_accuracy improved from 0.86413 to 0.86693, saving model to
best_model.h5
1329/1329 [==============================] - 9s 7ms/step - loss: 0.5047
- ce: 0.5047 - accuracy: 0.8287 - val_loss: 0.3912 - val_ce: 0.3912 - va
l_accuracy: 0.8669 - lr: 5.7653e-04
Epoch 35/50
1321/1329 [============================>.] - ETA: 0s - loss: 0.4957 - c
e: 0.4957 - accuracy: 0.8315
Epoch 35: val_accuracy improved from 0.86693 to 0.87293, saving model to
best_model.h5
1329/1329 [==============================] - 9s 7ms/step - loss: 0.4963
- ce: 0.4963 - accuracy: 0.8311 - val_loss: 0.3790 - val_ce: 0.3790 - va
l_accuracy: 0.8729 - lr: 5.5901e-04
Epoch 36/50
1326/1329 [============================>.] - ETA: 0s - loss: 0.4774 - c
e: 0.4774 - accuracy: 0.8376
Epoch 36: val_accuracy did not improve from 0.87293
1329/1329 [==============================] - 9s 7ms/step - loss: 0.4775
- ce: 0.4775 - accuracy: 0.8376 - val_loss: 0.3722 - val_ce: 0.3722 - va
l_accuracy: 0.8703 - lr: 3.8513e-04
Epoch 37/50
1324/1329 [============================>.] - ETA: 0s - loss: 0.5241 - c
e: 0.5241 - accuracy: 0.8212
Epoch 37: val_accuracy did not improve from 0.87293
1329/1329 [==============================] - 9s 7ms/step - loss: 0.5249
- ce: 0.5249 - accuracy: 0.8212 - val_loss: 0.4271 - val_ce: 0.4271 - va
l_accuracy: 0.8537 - lr: 9.9939e-04
Epoch 38/50
1321/1329 [============================>.] - ETA: 0s - loss: 0.4763 - c
e: 0.4763 - accuracy: 0.8373
Epoch 38: val_accuracy improved from 0.87293 to 0.87387, saving model to
best_model.h5
1329/1329 [==============================] - 9s 7ms/step - loss: 0.4760
- ce: 0.4760 - accuracy: 0.8373 - val_loss: 0.3587 - val_ce: 0.3587 - va
l_accuracy: 0.8739 - lr: 3.5214e-04
Epoch 39/50
1329/1329 [==============================] - ETA: 0s - loss: 0.4456 - c
e: 0.4456 - accuracy: 0.8479
Epoch 39: val_accuracy improved from 0.87387 to 0.88280, saving model to
best_model.h5
1329/1329 [==============================] - 9s 7ms/step - loss: 0.4456
- ce: 0.4456 - accuracy: 0.8479 - val_loss: 0.3326 - val_ce: 0.3326 - va
l_accuracy: 0.8828 - lr: 8.8288e-05
Epoch 40/50
1329/1329 [==============================] - ETA: 0s - loss: 0.4751 - c
e: 0.4751 - accuracy: 0.8404
Epoch 40: val_accuracy did not improve from 0.88280
1329/1329 [==============================] - 9s 7ms/step - loss: 0.4751
- ce: 0.4751 - accuracy: 0.8404 - val_loss: 0.3770 - val_ce: 0.3770 - va
l_accuracy: 0.8700 - lr: 5.3810e-04
Epoch 41/50
1325/1329 [============================>.] - ETA: 0s - loss: 0.4430 - c
e: 0.4430 - accuracy: 0.8502
Epoch 41: val_accuracy improved from 0.88280 to 0.88653, saving model to
best_model.h5
```

```
1329/1329 [==============================] - 9s 7ms/step - loss: 0.4428
- ce: 0.4428 - accuracy: 0.8503 - val_loss: 0.3373 - val_ce: 0.3373 - va
l_accuracy: 0.8865 - lr: 7.1154e-05
Epoch 42/50
1328/1329 [============================>.] - ETA: 0s - loss: 0.5049 - c
e: 0.5049 - accuracy: 0.8292
Epoch 42: val_accuracy did not improve from 0.88653
1329/1329 [==============================] - 9s 7ms/step - loss: 0.5050
- ce: 0.5050 - accuracy: 0.8292 - val_loss: 0.3982 - val_ce: 0.3982 - va
l_accuracy: 0.8625 - lr: 9.0997e-04
Epoch 43/50
1323/1329 [============================>.] - ETA: 0s - loss: 0.4585 - c
e: 0.4585 - accuracy: 0.8444
Epoch 43: val_accuracy did not improve from 0.88653
1329/1329 [==============================] - 9s 7ms/step - loss: 0.4586
- ce: 0.4586 - accuracy: 0.8443 - val_loss: 0.3748 - val_ce: 0.3748 - va
l_accuracy: 0.8744 - lr: 3.0913e-04
Epoch 44/50
1322/1329 [============================>.] - ETA: 0s - loss: 0.4935 - c
e: 0.4935 - accuracy: 0.8335
Epoch 44: val_accuracy did not improve from 0.88653
1329/1329 [==============================] - 9s 7ms/step - loss: 0.4930
- ce: 0.4930 - accuracy: 0.8338 - val_loss: 0.4305 - val_ce: 0.4305 - va
l_accuracy: 0.8521 - lr: 8.0221e-04
Epoch 45/50
1322/1329 [============================>.] - ETA: 0s - loss: 0.4470 - c
e: 0.4470 - accuracy: 0.8479
Epoch 45: val_accuracy did not improve from 0.88653
1329/1329 [==============================] - 9s 7ms/step - loss: 0.4472
- ce: 0.4472 - accuracy: 0.8479 - val_loss: 0.3398 - val_ce: 0.3398 - va
l_accuracy: 0.8836 - lr: 2.0688e-04
Epoch 46/50
1321/1329 [============================>.] - ETA: 0s - loss: 0.4369 - c
e: 0.4369 - accuracy: 0.8505
Epoch 46: val_accuracy improved from 0.88653 to 0.88973, saving model to
best_model.h5
1329/1329 [==============================] - 9s 7ms/step - loss: 0.4363
- ce: 0.4363 - accuracy: 0.8507 - val_loss: 0.3211 - val_ce: 0.3211 - va
l_accuracy: 0.8897 - lr: 2.3108e-04
Epoch 47/50
1327/1329 [============================>.] - ETA: 0s - loss: 0.4233 - c
e: 0.4233 - accuracy: 0.8575
Epoch 47: val_accuracy improved from 0.88973 to 0.89080, saving model to
best_model.h5
1329/1329 [==============================] - 9s 7ms/step - loss: 0.4234
- ce: 0.4234 - accuracy: 0.8575 - val_loss: 0.3228 - val_ce: 0.3228 - va
l_accuracy: 0.8908 - lr: 6.8688e-06
Epoch 48/50
1321/1329 [============================>.] - ETA: 0s - loss: 0.4612 - c
e: 0.4612 - accuracy: 0.8420
Epoch 48: val_accuracy did not improve from 0.89080
1329/1329 [==============================] - 9s 6ms/step - loss: 0.4606
- ce: 0.4606 - accuracy: 0.8423 - val_loss: 0.3863 - val_ce: 0.3863 - va
l_accuracy: 0.8697 - lr: 6.4382e-04
Epoch 49/50
1324/1329 [============================>.] - ETA: 0s - loss: 0.4441 - c
e: 0.4441 - accuracy: 0.8496
Epoch 49: val_accuracy did not improve from 0.89080
1329/1329 [==============================] - 9s 6ms/step - loss: 0.4440
- ce: 0.4440 - accuracy: 0.8495 - val_loss: 0.3274 - val_ce: 0.3274 - va
```

```
l_accuracy: 0.8881 - lr: 7.8052e-06
Epoch 50/50
1324/1329 [============================>.] - ETA: 0s - loss: 0.4550 - c
e: 0.4550 - accuracy: 0.8446
Epoch 50: val_accuracy did not improve from 0.89080
1329/1329 [==============================] - 9s 7ms/step - loss: 0.4547
- ce: 0.4547 - accuracy: 0.8447 - val_loss: 0.3189 - val_ce: 0.3189 - va
l_accuracy: 0.8903 - lr: 5.3536e-04

Test Results:
Test Loss: 0.4412
Test Cross-Entropy: 0.4412
Test Accuracy: 0.8553
```




```
🏆 COMPETITION SUMMARY:
📊 Parameters: 118,258 / 122,000 (96.9%)
🎯 Best Validation CE: 0.3189 (Epoch 50)
🎯 Best Validation Accuracy: 0.8908 (Epoch 47)
🎯 Best Validation Loss: 0.3189 (Epoch 50)
📈 Final Test CE: 0.4412
📈 Final Test Accuracy: 0.8553
```

In [24]:

```python
import tensorflow as tf
from tensorflow.keras import layers, models, optimizers
import matplotlib.pyplot as plt

# Simple, effective data augmentation
def simple_augmentation(image, label):
    image = tf.cast(image, tf.float32)
    # Only horizontal flip - keep it simple
    image = tf.image.random_flip_left_right(image)
    return image, label

# Create dataset pipeline
def create_dataset(images, labels, batch_size, is_training=True, validati
    dataset = tf.data.Dataset.from_tensor_slices((images, labels))

    if is_training:
        # Split training data for validation
        dataset_size = len(images)
        val_size = int(dataset_size * validation_split)
        train_size = dataset_size - val_size

        dataset = dataset.shuffle(10000, seed=42)
        train_dataset = dataset.take(train_size)
        val_dataset = dataset.skip(train_size)

        # Apply minimal augmentation only to training data
        train_dataset = train_dataset.cache()
        train_dataset = train_dataset.shuffle(5000, reshuffle_each_iterat
```

```
        train_dataset = train_dataset.map(simple_augmentation, num_parall
        train_dataset = train_dataset.batch(batch_size)
        train_dataset = train_dataset.prefetch(tf.data.AUTOTUNE)

        # Validation dataset without augmentation
        val_dataset = val_dataset.batch(batch_size)
        val_dataset = val_dataset.prefetch(tf.data.AUTOTUNE)

        return train_dataset, val_dataset
    else:
        dataset = dataset.batch(batch_size)
        dataset = dataset.prefetch(tf.data.AUTOTUNE)
        return dataset

# Create a parameter-efficient but effective model
def create_simple_effective_model():
    inputs = layers.Input(shape=(32, 32, 3))

    # Block 1 - smaller filters
    x = layers.Conv2D(24, (3, 3), padding='same', activation='relu')(inpu
    x = layers.BatchNormalization()(x)
    x = layers.Conv2D(24, (3, 3), padding='same', activation='relu')(x)
    x = layers.MaxPooling2D((2, 2))(x)
    x = layers.Dropout(0.25)(x)

    # Block 2 - moderate filters
    x = layers.Conv2D(48, (3, 3), padding='same', activation='relu')(x)
    x = layers.BatchNormalization()(x)
    x = layers.Conv2D(48, (3, 3), padding='same', activation='relu')(x)
    x = layers.MaxPooling2D((2, 2))(x)
    x = layers.Dropout(0.25)(x)

    # Block 3 - slightly smaller final block
    x = layers.Conv2D(68, (3, 3), padding='same', activation='relu')(x)
    x = layers.BatchNormalization()(x)
    x = layers.Conv2D(68, (3, 3), padding='same', activation='relu')(x)
    x = layers.Dropout(0.25)(x)

    # Global average pooling instead of flatten
    x = layers.GlobalAveragePooling2D()(x)

    # Slightly smaller dense layer
    x = layers.Dense(120, activation='relu')(x)  # 128->120
    x = layers.Dropout(0.5)(x)

    outputs = layers.Dense(10)(x)

    model = models.Model(inputs, outputs, name='competitive_model')
    return model

# Simple cosine decay
def cosine_decay_schedule(epoch, total_epochs=100):
    import math
    return 0.001 * 0.5 * (1 + math.cos(math.pi * epoch / total_epochs))

# Build the model
model = create_simple_effective_model()
model.build(input_shape=(None, 32, 32, 3))

# Compile with basic settings
```

```
optimizer = optimizers.Adam(learning_rate=0.001)
model.compile(
    optimizer=optimizer,
    loss=tf.keras.losses.SparseCategoricalCrossentropy(from_logits=True),
    metrics=[
        tf.keras.metrics.SparseCategoricalCrossentropy(from_logits=True,
        tf.keras.metrics.SparseCategoricalAccuracy(name='accuracy')
    ]
)

# Display model summary
model.summary()

# Calculate total parameters
total_params = model.count_params()
print(f"Total Parameters: {total_params:,}")
print(f"Parameter budget used: {total_params/122000*100:.1f}%")

# Simple callbacks focused on generalization
callbacks = [
    tf.keras.callbacks.LearningRateScheduler(cosine_decay_schedule),
    tf.keras.callbacks.EarlyStopping(
        monitor='val_accuracy',  # Monitor accuracy instead
        patience=10,
        restore_best_weights=True,
        mode='max',
        verbose=1
    ),
    tf.keras.callbacks.ModelCheckpoint(
        'best_model.h5',
        monitor='val_accuracy',
        save_best_only=True,
        mode='max',
        verbose=1
    )
]

# Load and preprocess CIFAR-10 dataset
(train_images, train_labels), (test_images, test_labels) = tf.keras.datas

# Normalize pixel values to be between 0 and 1
train_images = train_images.astype('float32') / 255.0
test_images = test_images.astype('float32') / 255.0

# Create datasets with larger validation split
batch_size = 32  # Smaller batch size
train_dataset, val_dataset = create_dataset(train_images, train_labels, b
test_dataset = create_dataset(test_images, test_labels, batch_size, is_tr

# Train the model with fewer epochs
history = model.fit(
    train_dataset,
    epochs=100,  # Much fewer epochs
    validation_data=val_dataset,
    callbacks=callbacks,
    verbose=1
)

# Evaluate on test set
test_results = model.evaluate(test_dataset, verbose=0)
```

```
print(f"\nTest Results:")
print(f"Test Loss: {test_results[0]:.4f}")
print(f"Test Cross-Entropy: {test_results[1]:.4f}")
print(f"Test Accuracy: {test_results[2]:.4f}")

# Plot training history with detailed CE information
plt.figure(figsize=(15, 5))

plt.subplot(1, 3, 1)
plt.plot(history.history['ce'], label='Train CE', alpha=0.8)
plt.plot(history.history['val_ce'], label='Val CE', alpha=0.8)

# Find and annotate lowest validation CE
min_val_ce = min(history.history['val_ce'])
min_val_ce_epoch = history.history['val_ce'].index(min_val_ce)
plt.annotate(f'Lowest Val CE: {min_val_ce:.4f}\nEpoch: {min_val_ce_epoch
             xy=(min_val_ce_epoch, min_val_ce),
             xytext=(min_val_ce_epoch + 5, min_val_ce + 0.1),
             arrowprops=dict(arrowstyle='->', color='red', alpha=0.7),
             bbox=dict(boxstyle="round,pad=0.3", facecolor='yellow', alph
             fontsize=10)

# Mark the point
plt.plot(min_val_ce_epoch, min_val_ce, 'ro', markersize=8, alpha=0.8)

plt.title(f'Cross Entropy (Best Val CE: {min_val_ce:.4f})')
plt.xlabel('Epoch')
plt.ylabel('Cross Entropy')
plt.legend()
plt.grid(True, alpha=0.3)

plt.subplot(1, 3, 2)
plt.plot(history.history['accuracy'], label='Train Acc', alpha=0.8)
plt.plot(history.history['val_accuracy'], label='Val Acc', alpha=0.8)

# Find and annotate highest validation accuracy
max_val_acc = max(history.history['val_accuracy'])
max_val_acc_epoch = history.history['val_accuracy'].index(max_val_acc)
plt.annotate(f'Best Val Acc: {max_val_acc:.4f}\nEpoch: {max_val_acc_epoch
             xy=(max_val_acc_epoch, max_val_acc),
             xytext=(max_val_acc_epoch + 5, max_val_acc - 0.05),
             arrowprops=dict(arrowstyle='->', color='green', alpha=0.7),
             bbox=dict(boxstyle="round,pad=0.3", facecolor='lightgreen',
             fontsize=10)

# Mark the point
plt.plot(max_val_acc_epoch, max_val_acc, 'go', markersize=8, alpha=0.8)

plt.title(f'Accuracy (Best Val Acc: {max_val_acc:.4f})')
plt.xlabel('Epoch')
plt.ylabel('Accuracy')
plt.legend()
plt.grid(True, alpha=0.3)

plt.subplot(1, 3, 3)
plt.plot(history.history['loss'], label='Train Loss', alpha=0.8)
plt.plot(history.history['val_loss'], label='Val Loss', alpha=0.8)

# Find and annotate lowest validation loss
min_val_loss = min(history.history['val_loss'])
```

```
min_val_loss_epoch = history.history['val_loss'].index(min_val_loss)
plt.annotate(f'Lowest Val Loss: {min_val_loss:.4f}\nEpoch: {min_val_loss_
             xy=(min_val_loss_epoch, min_val_loss),
             xytext=(min_val_loss_epoch + 5, min_val_loss + 0.1),
             arrowprops=dict(arrowstyle='->', color='blue', alpha=0.7),
             bbox=dict(boxstyle="round,pad=0.3", facecolor='lightblue', a
             fontsize=10)

# Mark the point
plt.plot(min_val_loss_epoch, min_val_loss, 'bo', markersize=8, alpha=0.8)

plt.title(f'Loss (Best Val Loss: {min_val_loss:.4f})')
plt.xlabel('Epoch')
plt.ylabel('Loss')
plt.legend()
plt.grid(True, alpha=0.3)

plt.tight_layout()
plt.show()

# Print summary of best results
print(f"\n🏆 COMPETITION SUMMARY:")
print(f"📊 Parameters: {total_params:,} / 122,000 ({total_params/122000*1
print(f"🎯 Best Validation CE: {min_val_ce:.4f} (Epoch {min_val_ce_epoch
print(f"🎯 Best Validation Accuracy: {max_val_acc:.4f} (Epoch {max_val_ac
print(f"🎯 Best Validation Loss: {min_val_loss:.4f} (Epoch {min_val_loss_
print(f"📈 Final Test CE: {test_results[1]:.4f}")
print(f"📈 Final Test Accuracy: {test_results[2]:.4f}")
```

```
Model: "competitive_model"
_________________________________________________________________
 Layer (type)                Output Shape              Param #   
=================================================================
 input_24 (InputLayer)       [(None, 32, 32, 3)]       0         
                                                                 
 conv2d_72 (Conv2D)          (None, 32, 32, 24)        672       
                                                                 
 batch_normalization_64 (Bat  (None, 32, 32, 24)       96        
 chNormalization)                                                
                                                                 
 conv2d_73 (Conv2D)          (None, 32, 32, 24)        5208      
                                                                 
 max_pooling2d_30 (MaxPoolin  (None, 16, 16, 24)       0         
 g2D)                                                            
                                                                 
 dropout_66 (Dropout)        (None, 16, 16, 24)        0         
                                                                 
 conv2d_74 (Conv2D)          (None, 16, 16, 48)        10416     
                                                                 
 batch_normalization_65 (Bat  (None, 16, 16, 48)       192       
 chNormalization)                                                
                                                                 
 conv2d_75 (Conv2D)          (None, 16, 16, 48)        20784     
                                                                 
 max_pooling2d_31 (MaxPoolin  (None, 8, 8, 48)         0         
 g2D)                                                            
                                                                 
 dropout_67 (Dropout)        (None, 8, 8, 48)          0         
                                                                 
 conv2d_76 (Conv2D)          (None, 8, 8, 68)          29444     
                                                                 
 batch_normalization_66 (Bat  (None, 8, 8, 68)         272       
 chNormalization)                                                
                                                                 
 conv2d_77 (Conv2D)          (None, 8, 8, 68)          41684     
                                                                 
 dropout_68 (Dropout)        (None, 8, 8, 68)          0         
                                                                 
 global_average_pooling2d_15  (None, 68)               0         
  (GlobalAveragePooling2D)                                       
                                                                 
 dense_32 (Dense)            (None, 120)               8280      
                                                                 
 dropout_69 (Dropout)        (None, 120)               0         
                                                                 
 dense_33 (Dense)            (None, 10)                1210      
                                                                 
=================================================================
Total params: 118,258
Trainable params: 117,978
Non-trainable params: 280
_________________________________________________________________
Total Parameters: 118,258
Parameter budget used: 96.9%
Epoch 1/100
1328/1329 [============================>.] - ETA: 0s - loss: 1.5131 - c
e: 1.5131 - accuracy: 0.4449
Epoch 1: val_accuracy improved from -inf to 0.43360, saving model to bes
t_model.h5
```

```
1329/1329 [==============================] - 10s 7ms/step - loss: 1.5132
- ce: 1.5132 - accuracy: 0.4448 - val_loss: 1.6150 - val_ce: 1.6150 - va
l_accuracy: 0.4336 - lr: 0.0010
Epoch 2/100
1329/1329 [==============================] - ETA: 0s - loss: 1.1680 - c
e: 1.1680 - accuracy: 0.5877
Epoch 2: val_accuracy improved from 0.43360 to 0.59627, saving model to
best_model.h5
1329/1329 [==============================] - 9s 7ms/step - loss: 1.1680
- ce: 1.1680 - accuracy: 0.5877 - val_loss: 1.2200 - val_ce: 1.2200 - va
l_accuracy: 0.5963 - lr: 0.0010
Epoch 3/100
1321/1329 [============================>.] - ETA: 0s - loss: 1.0455 - c
e: 1.0455 - accuracy: 0.6334
Epoch 3: val_accuracy improved from 0.59627 to 0.67560, saving model to
best_model.h5
1329/1329 [==============================] - 9s 7ms/step - loss: 1.0448
- ce: 1.0448 - accuracy: 0.6337 - val_loss: 0.9180 - val_ce: 0.9180 - va
l_accuracy: 0.6756 - lr: 1.0000e-03
Epoch 4/100
1327/1329 [============================>.] - ETA: 0s - loss: 0.9684 - c
e: 0.9684 - accuracy: 0.6612
Epoch 4: val_accuracy did not improve from 0.67560
1329/1329 [==============================] - 9s 7ms/step - loss: 0.9684
- ce: 0.9684 - accuracy: 0.6611 - val_loss: 1.3507 - val_ce: 1.3507 - va
l_accuracy: 0.5484 - lr: 1.0000e-03
Epoch 5/100
1321/1329 [============================>.] - ETA: 0s - loss: 0.9000 - c
e: 0.9000 - accuracy: 0.6869
Epoch 5: val_accuracy improved from 0.67560 to 0.67627, saving model to
best_model.h5
1329/1329 [==============================] - 9s 7ms/step - loss: 0.8996
- ce: 0.8996 - accuracy: 0.6872 - val_loss: 0.9402 - val_ce: 0.9402 - va
l_accuracy: 0.6763 - lr: 1.0000e-03
Epoch 6/100
1326/1329 [============================>.] - ETA: 0s - loss: 0.8503 - c
e: 0.8503 - accuracy: 0.7076
Epoch 6: val_accuracy improved from 0.67627 to 0.72453, saving model to
best_model.h5
1329/1329 [==============================] - 9s 7ms/step - loss: 0.8503
- ce: 0.8503 - accuracy: 0.7076 - val_loss: 0.8050 - val_ce: 0.8050 - va
l_accuracy: 0.7245 - lr: 1.0000e-03
Epoch 7/100
1326/1329 [============================>.] - ETA: 0s - loss: 0.8077 - c
e: 0.8077 - accuracy: 0.7220
Epoch 7: val_accuracy improved from 0.72453 to 0.76267, saving model to
best_model.h5
1329/1329 [==============================] - 9s 7ms/step - loss: 0.8075
- ce: 0.8075 - accuracy: 0.7220 - val_loss: 0.6854 - val_ce: 0.6854 - va
l_accuracy: 0.7627 - lr: 9.9958e-04
Epoch 8/100
1321/1329 [============================>.] - ETA: 0s - loss: 0.7271 - c
e: 0.7271 - accuracy: 0.7513
Epoch 8: val_accuracy improved from 0.76267 to 0.79573, saving model to
best_model.h5
1329/1329 [==============================] - 9s 7ms/step - loss: 0.7264
- ce: 0.7264 - accuracy: 0.7515 - val_loss: 0.5667 - val_ce: 0.5667 - va
l_accuracy: 0.7957 - lr: 1.8408e-05
Epoch 9/100
1324/1329 [============================>.] - ETA: 0s - loss: 0.6994 - c
```

```
e: 0.6994 - accuracy: 0.7611
Epoch 9: val_accuracy improved from 0.79573 to 0.80440, saving model to
best_model.h5
1329/1329 [==============================] - 9s 7ms/step - loss: 0.6995
- ce: 0.6995 - accuracy: 0.7611 - val_loss: 0.5487 - val_ce: 0.5487 - va
l_accuracy: 0.8044 - lr: 7.0725e-05
Epoch 10/100
1327/1329 [============================>.] - ETA: 0s - loss: 0.6871 - c
e: 0.6871 - accuracy: 0.7647
Epoch 10: val_accuracy improved from 0.80440 to 0.81307, saving model to
best_model.h5
1329/1329 [==============================] - 9s 7ms/step - loss: 0.6872
- ce: 0.6872 - accuracy: 0.7647 - val_loss: 0.5354 - val_ce: 0.5354 - va
l_accuracy: 0.8131 - lr: 2.0896e-04
Epoch 11/100
1324/1329 [============================>.] - ETA: 0s - loss: 0.6633 - c
e: 0.6633 - accuracy: 0.7721
Epoch 11: val_accuracy improved from 0.81307 to 0.81560, saving model to
best_model.h5
1329/1329 [==============================] - 9s 7ms/step - loss: 0.6632
- ce: 0.6632 - accuracy: 0.7722 - val_loss: 0.5292 - val_ce: 0.5292 - va
l_accuracy: 0.8156 - lr: 1.3365e-04
Epoch 12/100
1322/1329 [============================>.] - ETA: 0s - loss: 0.6715 - c
e: 0.6715 - accuracy: 0.7711
Epoch 12: val_accuracy did not improve from 0.81560
1329/1329 [==============================] - 9s 6ms/step - loss: 0.6714
- ce: 0.6714 - accuracy: 0.7712 - val_loss: 0.5352 - val_ce: 0.5352 - va
l_accuracy: 0.8133 - lr: 2.9049e-04
Epoch 13/100
1322/1329 [============================>.] - ETA: 0s - loss: 0.6394 - c
e: 0.6394 - accuracy: 0.7829
Epoch 13: val_accuracy improved from 0.81560 to 0.81733, saving model to
best_model.h5
1329/1329 [==============================] - 9s 7ms/step - loss: 0.6392
- ce: 0.6392 - accuracy: 0.7829 - val_loss: 0.5040 - val_ce: 0.5040 - va
l_accuracy: 0.8173 - lr: 7.9585e-05
Epoch 14/100
1321/1329 [============================>.] - ETA: 0s - loss: 0.7181 - c
e: 0.7181 - accuracy: 0.7536
Epoch 14: val_accuracy did not improve from 0.81733
1329/1329 [==============================] - 9s 7ms/step - loss: 0.7177
- ce: 0.7177 - accuracy: 0.7537 - val_loss: 0.5908 - val_ce: 0.5908 - va
l_accuracy: 0.7901 - lr: 8.7437e-04
Epoch 15/100
1325/1329 [============================>.] - ETA: 0s - loss: 0.6769 - c
e: 0.6769 - accuracy: 0.7709
Epoch 15: val_accuracy did not improve from 0.81733
1329/1329 [==============================] - 9s 7ms/step - loss: 0.6768
- ce: 0.6768 - accuracy: 0.7710 - val_loss: 0.5885 - val_ce: 0.5885 - va
l_accuracy: 0.7967 - lr: 5.3975e-04
Epoch 16/100
1329/1329 [==============================] - ETA: 0s - loss: 0.6664 - c
e: 0.6664 - accuracy: 0.7719
Epoch 16: val_accuracy did not improve from 0.81733
1329/1329 [==============================] - 9s 7ms/step - loss: 0.6664
- ce: 0.6664 - accuracy: 0.7719 - val_loss: 0.5646 - val_ce: 0.5646 - va
l_accuracy: 0.8012 - lr: 6.6456e-04
Epoch 17/100
1322/1329 [============================>.] - ETA: 0s - loss: 0.6835 - c
```

```
e: 0.6835 - accuracy: 0.7679
Epoch 17: val_accuracy did not improve from 0.81733
1329/1329 [==============================] - 9s 7ms/step - loss: 0.6834
- ce: 0.6834 - accuracy: 0.7680 - val_loss: 0.6396 - val_ce: 0.6396 - va
l_accuracy: 0.7773 - lr: 9.9973e-04
Epoch 18/100
1327/1329 [============================>.] - ETA: 0s - loss: 0.6077 - c
e: 0.6077 - accuracy: 0.7941
Epoch 18: val_accuracy did not improve from 0.81733
1329/1329 [==============================] - 9s 7ms/step - loss: 0.6076
- ce: 0.6076 - accuracy: 0.7941 - val_loss: 0.5282 - val_ce: 0.5282 - va
l_accuracy: 0.8168 - lr: 2.4941e-04
Epoch 19/100
1321/1329 [============================>.] - ETA: 0s - loss: 0.5897 - c
e: 0.5897 - accuracy: 0.8010
Epoch 19: val_accuracy improved from 0.81733 to 0.83267, saving model to
best_model.h5
1329/1329 [==============================] - 9s 7ms/step - loss: 0.5897
- ce: 0.5897 - accuracy: 0.8010 - val_loss: 0.4779 - val_ce: 0.4779 - va
l_accuracy: 0.8327 - lr: 2.7019e-04
Epoch 20/100
1323/1329 [============================>.] - ETA: 0s - loss: 0.6227 - c
e: 0.6227 - accuracy: 0.7882
Epoch 20: val_accuracy did not improve from 0.83267
1329/1329 [==============================] - 9s 7ms/step - loss: 0.6225
- ce: 0.6225 - accuracy: 0.7883 - val_loss: 0.4722 - val_ce: 0.4722 - va
l_accuracy: 0.8323 - lr: 6.9197e-04
Epoch 21/100
1325/1329 [============================>.] - ETA: 0s - loss: 0.5794 - c
e: 0.5794 - accuracy: 0.8009
Epoch 21: val_accuracy improved from 0.83267 to 0.84560, saving model to
best_model.h5
1329/1329 [==============================] - 9s 7ms/step - loss: 0.5792
- ce: 0.5792 - accuracy: 0.8010 - val_loss: 0.4513 - val_ce: 0.4513 - va
l_accuracy: 0.8456 - lr: 1.3757e-05
Epoch 22/100
1323/1329 [============================>.] - ETA: 0s - loss: 0.6430 - c
e: 0.6430 - accuracy: 0.7835
Epoch 22: val_accuracy did not improve from 0.84560
1329/1329 [==============================] - 9s 7ms/step - loss: 0.6432
- ce: 0.6432 - accuracy: 0.7835 - val_loss: 0.4849 - val_ce: 0.4849 - va
l_accuracy: 0.8365 - lr: 9.9646e-04
Epoch 23/100
1323/1329 [============================>.] - ETA: 0s - loss: 0.6220 - c
e: 0.6220 - accuracy: 0.7876
Epoch 23: val_accuracy did not improve from 0.84560
1329/1329 [==============================] - 9s 7ms/step - loss: 0.6219
- ce: 0.6219 - accuracy: 0.7878 - val_loss: 0.5419 - val_ce: 0.5419 - va
l_accuracy: 0.8159 - lr: 9.0587e-04
Epoch 24/100
1328/1329 [============================>.] - ETA: 0s - loss: 0.6183 - c
e: 0.6183 - accuracy: 0.7904
Epoch 24: val_accuracy did not improve from 0.84560
1329/1329 [==============================] - 9s 7ms/step - loss: 0.6183
- ce: 0.6183 - accuracy: 0.7904 - val_loss: 0.5742 - val_ce: 0.5742 - va
l_accuracy: 0.8029 - lr: 9.9776e-04
Epoch 25/100
1323/1329 [============================>.] - ETA: 0s - loss: 0.6104 - c
e: 0.6104 - accuracy: 0.7940
Epoch 25: val_accuracy did not improve from 0.84560
```

```
1329/1329 [==============================] - 9s 7ms/step - loss: 0.6100
- ce: 0.6100 - accuracy: 0.7941 - val_loss: 0.4998 - val_ce: 0.4998 - va
l_accuracy: 0.8293 - lr: 9.9978e-04
Epoch 26/100
1325/1329 [============================>.] - ETA: 0s - loss: 0.5552 - c
e: 0.5552 - accuracy: 0.8127
Epoch 26: val_accuracy did not improve from 0.84560
1329/1329 [==============================] - 9s 7ms/step - loss: 0.5547
- ce: 0.5547 - accuracy: 0.8128 - val_loss: 0.4478 - val_ce: 0.4478 - va
l_accuracy: 0.8444 - lr: 5.3757e-04
Epoch 27/100
1324/1329 [============================>.] - ETA: 0s - loss: 0.5840 - c
e: 0.5840 - accuracy: 0.8028
Epoch 27: val_accuracy did not improve from 0.84560
1329/1329 [==============================] - 9s 7ms/step - loss: 0.5838
- ce: 0.5838 - accuracy: 0.8028 - val_loss: 0.4517 - val_ce: 0.4517 - va
l_accuracy: 0.8451 - lr: 9.5489e-04
Epoch 28/100
1322/1329 [============================>.] - ETA: 0s - loss: 0.5287 - c
e: 0.5287 - accuracy: 0.8201
Epoch 28: val_accuracy improved from 0.84560 to 0.85653, saving model to
best_model.h5
1329/1329 [==============================] - 9s 7ms/step - loss: 0.5285
- ce: 0.5285 - accuracy: 0.8202 - val_loss: 0.4098 - val_ce: 0.4098 - va
l_accuracy: 0.8565 - lr: 3.4819e-04
Epoch 29/100
1324/1329 [============================>.] - ETA: 0s - loss: 0.4975 - c
e: 0.4975 - accuracy: 0.8310
Epoch 29: val_accuracy improved from 0.85653 to 0.86653, saving model to
best_model.h5
1329/1329 [==============================] - 9s 7ms/step - loss: 0.4974
- ce: 0.4974 - accuracy: 0.8310 - val_loss: 0.3946 - val_ce: 0.3946 - va
l_accuracy: 0.8665 - lr: 1.6462e-04
Epoch 30/100
1325/1329 [============================>.] - ETA: 0s - loss: 0.5203 - c
e: 0.5203 - accuracy: 0.8240
Epoch 30: val_accuracy improved from 0.86653 to 0.87093, saving model to
best_model.h5
1329/1329 [==============================] - 9s 7ms/step - loss: 0.5203
- ce: 0.5203 - accuracy: 0.8240 - val_loss: 0.3794 - val_ce: 0.3794 - va
l_accuracy: 0.8709 - lr: 4.6585e-04
Epoch 31/100
1322/1329 [============================>.] - ETA: 0s - loss: 0.5644 - c
e: 0.5644 - accuracy: 0.8100
Epoch 31: val_accuracy did not improve from 0.87093
1329/1329 [==============================] - 9s 7ms/step - loss: 0.5640
- ce: 0.5640 - accuracy: 0.8102 - val_loss: 0.4910 - val_ce: 0.4910 - va
l_accuracy: 0.8339 - lr: 9.4167e-04
Epoch 32/100
1329/1329 [==============================] - ETA: 0s - loss: 0.5405 - c
e: 0.5405 - accuracy: 0.8168
Epoch 32: val_accuracy did not improve from 0.87093
1329/1329 [==============================] - 9s 7ms/step - loss: 0.5405
- ce: 0.5405 - accuracy: 0.8168 - val_loss: 0.3963 - val_ce: 0.3963 - va
l_accuracy: 0.8643 - lr: 7.6562e-04
Epoch 33/100
1325/1329 [============================>.] - ETA: 0s - loss: 0.5556 - c
e: 0.5556 - accuracy: 0.8107
Epoch 33: val_accuracy did not improve from 0.87093
1329/1329 [==============================] - 9s 7ms/step - loss: 0.5552
```

```
- ce: 0.5552 - accuracy: 0.8109 - val_loss: 0.4806 - val_ce: 0.4806 - va
l_accuracy: 0.8367 - lr: 9.6324e-04
Epoch 34/100
1329/1329 [==============================] - ETA: 0s - loss: 0.5102 - c
e: 0.5102 - accuracy: 0.8278
Epoch 34: val_accuracy did not improve from 0.87093
1329/1329 [==============================] - 9s 7ms/step - loss: 0.5102
- ce: 0.5102 - accuracy: 0.8278 - val_loss: 0.4366 - val_ce: 0.4366 - va
l_accuracy: 0.8501 - lr: 5.7653e-04
Epoch 35/100
1325/1329 [============================>.] - ETA: 0s - loss: 0.5025 - c
e: 0.5025 - accuracy: 0.8289
Epoch 35: val_accuracy did not improve from 0.87093
1329/1329 [==============================] - 9s 7ms/step - loss: 0.5024
- ce: 0.5024 - accuracy: 0.8290 - val_loss: 0.3727 - val_ce: 0.3727 - va
l_accuracy: 0.8707 - lr: 5.5901e-04
Epoch 36/100
1329/1329 [==============================] - ETA: 0s - loss: 0.4821 - c
e: 0.4821 - accuracy: 0.8359
Epoch 36: val_accuracy did not improve from 0.87093
1329/1329 [==============================] - 9s 7ms/step - loss: 0.4821
- ce: 0.4821 - accuracy: 0.8359 - val_loss: 0.3738 - val_ce: 0.3738 - va
l_accuracy: 0.8709 - lr: 3.8513e-04
Epoch 37/100
1323/1329 [============================>.] - ETA: 0s - loss: 0.5313 - c
e: 0.5313 - accuracy: 0.8197
Epoch 37: val_accuracy did not improve from 0.87093
1329/1329 [==============================] - 9s 7ms/step - loss: 0.5314
- ce: 0.5314 - accuracy: 0.8196 - val_loss: 0.4476 - val_ce: 0.4476 - va
l_accuracy: 0.8488 - lr: 9.9939e-04
Epoch 38/100
1328/1329 [============================>.] - ETA: 0s - loss: 0.4857 - c
e: 0.4857 - accuracy: 0.8365
Epoch 38: val_accuracy improved from 0.87093 to 0.87480, saving model to
best_model.h5
1329/1329 [==============================] - 9s 7ms/step - loss: 0.4857
- ce: 0.4857 - accuracy: 0.8365 - val_loss: 0.3610 - val_ce: 0.3610 - va
l_accuracy: 0.8748 - lr: 3.5214e-04
Epoch 39/100
1325/1329 [============================>.] - ETA: 0s - loss: 0.4496 - c
e: 0.4496 - accuracy: 0.8463
Epoch 39: val_accuracy improved from 0.87480 to 0.88533, saving model to
best_model.h5
1329/1329 [==============================] - 9s 7ms/step - loss: 0.4496
- ce: 0.4496 - accuracy: 0.8463 - val_loss: 0.3401 - val_ce: 0.3401 - va
l_accuracy: 0.8853 - lr: 8.8288e-05
Epoch 40/100
1320/1329 [============================>.] - ETA: 0s - loss: 0.4857 - c
e: 0.4857 - accuracy: 0.8348
Epoch 40: val_accuracy did not improve from 0.88533
1329/1329 [==============================] - 9s 7ms/step - loss: 0.4856
- ce: 0.4856 - accuracy: 0.8347 - val_loss: 0.3776 - val_ce: 0.3776 - va
l_accuracy: 0.8669 - lr: 5.3810e-04
Epoch 41/100
1329/1329 [==============================] - ETA: 0s - loss: 0.4484 - c
e: 0.4484 - accuracy: 0.8471
Epoch 41: val_accuracy improved from 0.88533 to 0.88773, saving model to
best_model.h5
1329/1329 [==============================] - 9s 7ms/step - loss: 0.4484
- ce: 0.4484 - accuracy: 0.8471 - val_loss: 0.3435 - val_ce: 0.3435 - va
```

```
l_accuracy: 0.8877 - lr: 7.1154e-05
Epoch 42/100
1323/1329 [============================>.] - ETA: 0s - loss: 0.5136 - c
e: 0.5136 - accuracy: 0.8260
Epoch 42: val_accuracy did not improve from 0.88773
1329/1329 [==============================] - 9s 6ms/step - loss: 0.5138
- ce: 0.5138 - accuracy: 0.8260 - val_loss: 0.4268 - val_ce: 0.4268 - va
l_accuracy: 0.8473 - lr: 9.0997e-04
Epoch 43/100
1327/1329 [============================>.] - ETA: 0s - loss: 0.4641 - c
e: 0.4641 - accuracy: 0.8419
Epoch 43: val_accuracy did not improve from 0.88773
1329/1329 [==============================] - 9s 7ms/step - loss: 0.4642
- ce: 0.4642 - accuracy: 0.8418 - val_loss: 0.3720 - val_ce: 0.3720 - va
l_accuracy: 0.8773 - lr: 3.0913e-04
Epoch 44/100
1328/1329 [============================>.] - ETA: 0s - loss: 0.4989 - c
e: 0.4989 - accuracy: 0.8296
Epoch 44: val_accuracy did not improve from 0.88773
1329/1329 [==============================] - 9s 7ms/step - loss: 0.4990
- ce: 0.4990 - accuracy: 0.8296 - val_loss: 0.3961 - val_ce: 0.3961 - va
l_accuracy: 0.8627 - lr: 8.0221e-04
Epoch 45/100
1324/1329 [============================>.] - ETA: 0s - loss: 0.4436 - c
e: 0.4436 - accuracy: 0.8505
Epoch 45: val_accuracy did not improve from 0.88773
1329/1329 [==============================] - 8s 6ms/step - loss: 0.4440
- ce: 0.4440 - accuracy: 0.8504 - val_loss: 0.3462 - val_ce: 0.3462 - va
l_accuracy: 0.8855 - lr: 2.0688e-04
Epoch 46/100
1327/1329 [============================>.] - ETA: 0s - loss: 0.4380 - c
e: 0.4380 - accuracy: 0.8503
Epoch 46: val_accuracy improved from 0.88773 to 0.89307, saving model to
best_model.h5
1329/1329 [==============================] - 9s 7ms/step - loss: 0.4383
- ce: 0.4383 - accuracy: 0.8503 - val_loss: 0.3246 - val_ce: 0.3246 - va
l_accuracy: 0.8931 - lr: 2.3108e-04
Epoch 47/100
1325/1329 [============================>.] - ETA: 0s - loss: 0.4231 - c
e: 0.4231 - accuracy: 0.8565
Epoch 47: val_accuracy did not improve from 0.89307
1329/1329 [==============================] - 9s 7ms/step - loss: 0.4232
- ce: 0.4232 - accuracy: 0.8564 - val_loss: 0.3307 - val_ce: 0.3307 - va
l_accuracy: 0.8907 - lr: 6.8688e-06
Epoch 48/100
1323/1329 [============================>.] - ETA: 0s - loss: 0.4689 - c
e: 0.4689 - accuracy: 0.8407
Epoch 48: val_accuracy did not improve from 0.89307
1329/1329 [==============================] - 9s 7ms/step - loss: 0.4691
- ce: 0.4691 - accuracy: 0.8407 - val_loss: 0.3854 - val_ce: 0.3854 - va
l_accuracy: 0.8652 - lr: 6.4382e-04
Epoch 49/100
1324/1329 [============================>.] - ETA: 0s - loss: 0.4434 - c
e: 0.4434 - accuracy: 0.8478
Epoch 49: val_accuracy did not improve from 0.89307
1329/1329 [==============================] - 9s 7ms/step - loss: 0.4437
- ce: 0.4437 - accuracy: 0.8477 - val_loss: 0.3504 - val_ce: 0.3504 - va
l_accuracy: 0.8775 - lr: 7.8052e-06
Epoch 50/100
1322/1329 [============================>.] - ETA: 0s - loss: 0.4602 - c
```

```
e: 0.4602 - accuracy: 0.8424
Epoch 50: val_accuracy did not improve from 0.89307
1329/1329 [==============================] - 9s 7ms/step - loss: 0.4599
- ce: 0.4599 - accuracy: 0.8424 - val_loss: 0.3595 - val_ce: 0.3595 - va
l_accuracy: 0.8751 - lr: 5.3536e-04
Epoch 51/100
1325/1329 [============================>.] - ETA: 0s - loss: 0.4703 - c
e: 0.4703 - accuracy: 0.8397
Epoch 51: val_accuracy did not improve from 0.89307
1329/1329 [==============================] - 9s 7ms/step - loss: 0.4702
- ce: 0.4702 - accuracy: 0.8398 - val_loss: 0.3646 - val_ce: 0.3646 - va
l_accuracy: 0.8752 - lr: 6.7547e-04
Epoch 52/100
1329/1329 [==============================] - ETA: 0s - loss: 0.4514 - c
e: 0.4514 - accuracy: 0.8454
Epoch 52: val_accuracy did not improve from 0.89307
1329/1329 [==============================] - 9s 7ms/step - loss: 0.4514
- ce: 0.4514 - accuracy: 0.8454 - val_loss: 0.3376 - val_ce: 0.3376 - va
l_accuracy: 0.8835 - lr: 5.1833e-04
Epoch 53/100
1325/1329 [============================>.] - ETA: 0s - loss: 0.4630 - c
e: 0.4630 - accuracy: 0.8423
Epoch 53: val_accuracy did not improve from 0.89307
1329/1329 [==============================] - 9s 6ms/step - loss: 0.4627
- ce: 0.4627 - accuracy: 0.8425 - val_loss: 0.3465 - val_ce: 0.3465 - va
l_accuracy: 0.8843 - lr: 6.8508e-04
Epoch 54/100
1324/1329 [============================>.] - ETA: 0s - loss: 0.4264 - c
e: 0.4264 - accuracy: 0.8537
Epoch 54: val_accuracy improved from 0.89307 to 0.89347, saving model to
best_model.h5
1329/1329 [==============================] - 9s 7ms/step - loss: 0.4262
- ce: 0.4262 - accuracy: 0.8536 - val_loss: 0.3230 - val_ce: 0.3230 - va
l_accuracy: 0.8935 - lr: 2.2975e-04
Epoch 55/100
1322/1329 [============================>.] - ETA: 0s - loss: 0.4305 - c
e: 0.4305 - accuracy: 0.8530
Epoch 55: val_accuracy did not improve from 0.89347
1329/1329 [==============================] - 9s 7ms/step - loss: 0.4310
- ce: 0.4310 - accuracy: 0.8529 - val_loss: 0.3219 - val_ce: 0.3219 - va
l_accuracy: 0.8929 - lr: 3.7783e-04
Epoch 56/100
1323/1329 [============================>.] - ETA: 0s - loss: 0.4887 - c
e: 0.4887 - accuracy: 0.8350
Epoch 56: val_accuracy did not improve from 0.89347
1329/1329 [==============================] - 9s 7ms/step - loss: 0.4885
- ce: 0.4885 - accuracy: 0.8350 - val_loss: 0.3944 - val_ce: 0.3944 - va
l_accuracy: 0.8636 - lr: 9.9988e-04
Epoch 57/100
1326/1329 [============================>.] - ETA: 0s - loss: 0.4447 - c
e: 0.4447 - accuracy: 0.8487
Epoch 57: val_accuracy did not improve from 0.89347
1329/1329 [==============================] - 9s 7ms/step - loss: 0.4449
- ce: 0.4449 - accuracy: 0.8486 - val_loss: 0.3356 - val_ce: 0.3356 - va
l_accuracy: 0.8843 - lr: 2.3470e-05
Epoch 58/100
1326/1329 [============================>.] - ETA: 0s - loss: 0.4311 - c
e: 0.4311 - accuracy: 0.8547
Epoch 58: val_accuracy did not improve from 0.89347
1329/1329 [==============================] - 9s 7ms/step - loss: 0.4310
```

```
- ce: 0.4310 - accuracy: 0.8548 - val_loss: 0.3361 - val_ce: 0.3361 - va
l_accuracy: 0.8839 - lr: 6.6471e-07
Epoch 59/100
1320/1329 [============================>.] - ETA: 0s - loss: 0.4850 - c
e: 0.4850 - accuracy: 0.8359
Epoch 59: val_accuracy did not improve from 0.89347
1329/1329 [==============================] - 9s 7ms/step - loss: 0.4845
- ce: 0.4845 - accuracy: 0.8360 - val_loss: 0.3735 - val_ce: 0.3735 - va
l_accuracy: 0.8747 - lr: 8.6346e-04
Epoch 60/100
1319/1329 [============================>.] - ETA: 0s - loss: 0.4855 - c
e: 0.4855 - accuracy: 0.8343
Epoch 60: val_accuracy did not improve from 0.89347
1329/1329 [==============================] - 9s 7ms/step - loss: 0.4852
- ce: 0.4852 - accuracy: 0.8345 - val_loss: 0.3721 - val_ce: 0.3721 - va
l_accuracy: 0.8713 - lr: 9.8839e-04
Epoch 61/100
1321/1329 [============================>.] - ETA: 0s - loss: 0.4308 - c
e: 0.4308 - accuracy: 0.8547
Epoch 61: val_accuracy did not improve from 0.89347
1329/1329 [==============================] - 9s 7ms/step - loss: 0.4308
- ce: 0.4308 - accuracy: 0.8546 - val_loss: 0.3211 - val_ce: 0.3211 - va
l_accuracy: 0.8917 - lr: 1.3859e-04
Epoch 62/100
1323/1329 [============================>.] - ETA: 0s - loss: 0.4097 - c
e: 0.4097 - accuracy: 0.8602
Epoch 62: val_accuracy improved from 0.89347 to 0.89387, saving model to
best_model.h5
1329/1329 [==============================] - 9s 7ms/step - loss: 0.4097
- ce: 0.4097 - accuracy: 0.8602 - val_loss: 0.3123 - val_ce: 0.3123 - va
l_accuracy: 0.8939 - lr: 1.3981e-05
Epoch 63/100
1324/1329 [============================>.] - ETA: 0s - loss: 0.4407 - c
e: 0.4407 - accuracy: 0.8501
Epoch 63: val_accuracy did not improve from 0.89387
1329/1329 [==============================] - 9s 7ms/step - loss: 0.4405
- ce: 0.4405 - accuracy: 0.8502 - val_loss: 0.3413 - val_ce: 0.3413 - va
l_accuracy: 0.8865 - lr: 5.0618e-04
Epoch 64/100
1324/1329 [============================>.] - ETA: 0s - loss: 0.4430 - c
e: 0.4430 - accuracy: 0.8504
Epoch 64: val_accuracy did not improve from 0.89387
1329/1329 [==============================] - 9s 7ms/step - loss: 0.4431
- ce: 0.4431 - accuracy: 0.8505 - val_loss: 0.3456 - val_ce: 0.3456 - va
l_accuracy: 0.8836 - lr: 6.1016e-04
Epoch 65/100
1329/1329 [==============================] - ETA: 0s - loss: 0.4329 - c
e: 0.4329 - accuracy: 0.8533
Epoch 65: val_accuracy did not improve from 0.89387
1329/1329 [==============================] - 9s 6ms/step - loss: 0.4329
- ce: 0.4329 - accuracy: 0.8533 - val_loss: 0.3260 - val_ce: 0.3260 - va
l_accuracy: 0.8923 - lr: 4.9950e-04
Epoch 66/100
1326/1329 [============================>.] - ETA: 0s - loss: 0.4033 - c
e: 0.4033 - accuracy: 0.8636
Epoch 66: val_accuracy improved from 0.89387 to 0.89493, saving model to
best_model.h5
1329/1329 [==============================] - 9s 7ms/step - loss: 0.4033
- ce: 0.4033 - accuracy: 0.8636 - val_loss: 0.3214 - val_ce: 0.3214 - va
l_accuracy: 0.8949 - lr: 2.4433e-04
```

```
Epoch 67/100
1323/1329 [============================>.] - ETA: 0s - loss: 0.3954 - c
e: 0.3954 - accuracy: 0.8652
Epoch 67: val_accuracy did not improve from 0.89493
1329/1329 [==============================] - 9s 7ms/step - loss: 0.3955
- ce: 0.3955 - accuracy: 0.8650 - val_loss: 0.3100 - val_ce: 0.3100 - va
l_accuracy: 0.8937 - lr: 2.2615e-04
Epoch 68/100
1322/1329 [============================>.] - ETA: 0s - loss: 0.4532 - c
e: 0.4532 - accuracy: 0.8466
Epoch 68: val_accuracy did not improve from 0.89493
1329/1329 [==============================] - 9s 7ms/step - loss: 0.4533
- ce: 0.4533 - accuracy: 0.8465 - val_loss: 0.3299 - val_ce: 0.3299 - va
l_accuracy: 0.8855 - lr: 8.7259e-04
Epoch 69/100
1324/1329 [============================>.] - ETA: 0s - loss: 0.4076 - c
e: 0.4076 - accuracy: 0.8601
Epoch 69: val_accuracy improved from 0.89493 to 0.90107, saving model to
best_model.h5
1329/1329 [==============================] - 9s 7ms/step - loss: 0.4073
- ce: 0.4073 - accuracy: 0.8601 - val_loss: 0.2963 - val_ce: 0.2963 - va
l_accuracy: 0.9011 - lr: 7.7386e-05
Epoch 70/100
1329/1329 [==============================] - ETA: 0s - loss: 0.4127 - c
e: 0.4127 - accuracy: 0.8604
Epoch 70: val_accuracy did not improve from 0.90107
1329/1329 [==============================] - 9s 7ms/step - loss: 0.4127
- ce: 0.4127 - accuracy: 0.8604 - val_loss: 0.3228 - val_ce: 0.3228 - va
l_accuracy: 0.8923 - lr: 3.6795e-04
Epoch 71/100
1319/1329 [============================>.] - ETA: 0s - loss: 0.3993 - c
e: 0.3993 - accuracy: 0.8631
Epoch 71: val_accuracy improved from 0.90107 to 0.90440, saving model to
best_model.h5
1329/1329 [==============================] - 9s 7ms/step - loss: 0.3991
- ce: 0.3991 - accuracy: 0.8633 - val_loss: 0.2940 - val_ce: 0.2940 - va
l_accuracy: 0.9044 - lr: 2.5215e-04
Epoch 72/100
1329/1329 [==============================] - ETA: 0s - loss: 0.3958 - c
e: 0.3958 - accuracy: 0.8654
Epoch 72: val_accuracy did not improve from 0.90440
1329/1329 [==============================] - 9s 7ms/step - loss: 0.3958
- ce: 0.3958 - accuracy: 0.8654 - val_loss: 0.3143 - val_ce: 0.3143 - va
l_accuracy: 0.8933 - lr: 3.4776e-04
Epoch 73/100
1323/1329 [============================>.] - ETA: 0s - loss: 0.4195 - c
e: 0.4195 - accuracy: 0.8548
Epoch 73: val_accuracy did not improve from 0.90440
1329/1329 [==============================] - 9s 7ms/step - loss: 0.4198
- ce: 0.4198 - accuracy: 0.8549 - val_loss: 0.3275 - val_ce: 0.3275 - va
l_accuracy: 0.8889 - lr: 6.3576e-04
Epoch 74/100
1329/1329 [==============================] - ETA: 0s - loss: 0.3914 - c
e: 0.3914 - accuracy: 0.8662
Epoch 74: val_accuracy did not improve from 0.90440
1329/1329 [==============================] - 9s 7ms/step - loss: 0.3914
- ce: 0.3914 - accuracy: 0.8662 - val_loss: 0.2927 - val_ce: 0.2927 - va
l_accuracy: 0.8987 - lr: 1.3758e-04
Epoch 75/100
1321/1329 [============================>.] - ETA: 0s - loss: 0.4313 - c
```

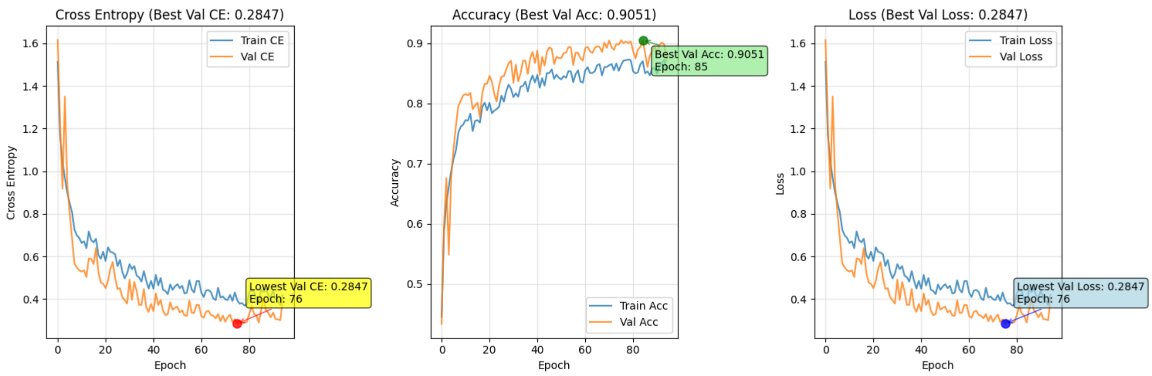
```
e: 0.4313 - accuracy: 0.8539
Epoch 75: val_accuracy did not improve from 0.90440
1329/1329 [==============================] - 9s 6ms/step - loss: 0.4314
- ce: 0.4314 - accuracy: 0.8539 - val_loss: 0.3055 - val_ce: 0.3055 - va
l_accuracy: 0.8976 - lr: 7.2000e-04
Epoch 76/100
1323/1329 [============================>.] - ETA: 0s - loss: 0.3950 - c
e: 0.3950 - accuracy: 0.8641
Epoch 76: val_accuracy improved from 0.90440 to 0.90493, saving model to
best_model.h5
1329/1329 [==============================] - 9s 7ms/step - loss: 0.3949
- ce: 0.3949 - accuracy: 0.8641 - val_loss: 0.2847 - val_ce: 0.2847 - va
l_accuracy: 0.9049 - lr: 1.4722e-04
Epoch 77/100
1322/1329 [============================>.] - ETA: 0s - loss: 0.3789 - c
e: 0.3789 - accuracy: 0.8709
Epoch 77: val_accuracy did not improve from 0.90493
1329/1329 [==============================] - 9s 7ms/step - loss: 0.3786
- ce: 0.3786 - accuracy: 0.8710 - val_loss: 0.2994 - val_ce: 0.2994 - va
l_accuracy: 0.9000 - lr: 1.0484e-06
Epoch 78/100
1329/1329 [==============================] - ETA: 0s - loss: 0.3811 - c
e: 0.3811 - accuracy: 0.8717
Epoch 78: val_accuracy did not improve from 0.90493
1329/1329 [==============================] - 9s 7ms/step - loss: 0.3811
- ce: 0.3811 - accuracy: 0.8717 - val_loss: 0.2853 - val_ce: 0.2853 - va
l_accuracy: 0.9029 - lr: 2.0388e-04
Epoch 79/100
1327/1329 [============================>.] - ETA: 0s - loss: 0.3695 - c
e: 0.3695 - accuracy: 0.8729
Epoch 79: val_accuracy did not improve from 0.90493
1329/1329 [==============================] - 9s 6ms/step - loss: 0.3697
- ce: 0.3697 - accuracy: 0.8729 - val_loss: 0.2904 - val_ce: 0.2904 - va
l_accuracy: 0.9001 - lr: 5.2597e-06
Epoch 80/100
1324/1329 [============================>.] - ETA: 0s - loss: 0.3732 - c
e: 0.3732 - accuracy: 0.8717
Epoch 80: val_accuracy did not improve from 0.90493
1329/1329 [==============================] - 9s 7ms/step - loss: 0.3736
- ce: 0.3736 - accuracy: 0.8716 - val_loss: 0.2935 - val_ce: 0.2935 - va
l_accuracy: 0.9029 - lr: 1.1849e-04
Epoch 81/100
1329/1329 [==============================] - ETA: 0s - loss: 0.4305 - c
e: 0.4305 - accuracy: 0.8533
Epoch 81: val_accuracy did not improve from 0.90493
1329/1329 [==============================] - 9s 7ms/step - loss: 0.4305
- ce: 0.4305 - accuracy: 0.8533 - val_loss: 0.3302 - val_ce: 0.3302 - va
l_accuracy: 0.8871 - lr: 8.2062e-04
Epoch 82/100
1325/1329 [============================>.] - ETA: 0s - loss: 0.4429 - c
e: 0.4429 - accuracy: 0.8496
Epoch 82: val_accuracy did not improve from 0.90493
1329/1329 [==============================] - 9s 7ms/step - loss: 0.4429
- ce: 0.4429 - accuracy: 0.8497 - val_loss: 0.3677 - val_ce: 0.3677 - va
l_accuracy: 0.8745 - lr: 8.9342e-04
Epoch 83/100
1320/1329 [============================>.] - ETA: 0s - loss: 0.4357 - c
e: 0.4357 - accuracy: 0.8514
Epoch 83: val_accuracy did not improve from 0.90493
1329/1329 [==============================] - 9s 6ms/step - loss: 0.4357
```

```
- ce: 0.4357 - accuracy: 0.8516 - val_loss: 0.3322 - val_ce: 0.3322 - va
l_accuracy: 0.8896 - lr: 8.1578e-04
Epoch 84/100
1325/1329 [============================>.] - ETA: 0s - loss: 0.3937 - c
e: 0.3937 - accuracy: 0.8646
Epoch 84: val_accuracy did not improve from 0.90493
1329/1329 [==============================] - 9s 7ms/step - loss: 0.3940
- ce: 0.3940 - accuracy: 0.8645 - val_loss: 0.3224 - val_ce: 0.3224 - va
l_accuracy: 0.8937 - lr: 2.6885e-04
Epoch 85/100
1321/1329 [============================>.] - ETA: 0s - loss: 0.3796 - c
e: 0.3796 - accuracy: 0.8700
Epoch 85: val_accuracy improved from 0.90493 to 0.90507, saving model to
best_model.h5
1329/1329 [==============================] - 9s 7ms/step - loss: 0.3795
- ce: 0.3795 - accuracy: 0.8701 - val_loss: 0.2902 - val_ce: 0.2902 - va
l_accuracy: 0.9051 - lr: 6.5294e-06
Epoch 86/100
1323/1329 [============================>.] - ETA: 0s - loss: 0.4467 - c
e: 0.4467 - accuracy: 0.8499
Epoch 86: val_accuracy did not improve from 0.90507
1329/1329 [==============================] - 9s 7ms/step - loss: 0.4465
- ce: 0.4465 - accuracy: 0.8500 - val_loss: 0.3511 - val_ce: 0.3511 - va
l_accuracy: 0.8837 - lr: 9.9010e-04
Epoch 87/100
1328/1329 [============================>.] - ETA: 0s - loss: 0.4330 - c
e: 0.4330 - accuracy: 0.8530
Epoch 87: val_accuracy did not improve from 0.90507
1329/1329 [==============================] - 9s 7ms/step - loss: 0.4330
- ce: 0.4330 - accuracy: 0.8530 - val_loss: 0.4028 - val_ce: 0.4028 - va
l_accuracy: 0.8603 - lr: 8.2706e-04
Epoch 88/100
1324/1329 [============================>.] - ETA: 0s - loss: 0.4481 - c
e: 0.4481 - accuracy: 0.8466
Epoch 88: val_accuracy did not improve from 0.90507
1329/1329 [==============================] - 9s 7ms/step - loss: 0.4486
- ce: 0.4486 - accuracy: 0.8464 - val_loss: 0.3486 - val_ce: 0.3486 - va
l_accuracy: 0.8795 - lr: 9.6985e-04
Epoch 89/100
1321/1329 [============================>.] - ETA: 0s - loss: 0.4168 - c
e: 0.4168 - accuracy: 0.8584
Epoch 89: val_accuracy did not improve from 0.90507
1329/1329 [==============================] - 9s 6ms/step - loss: 0.4167
- ce: 0.4167 - accuracy: 0.8584 - val_loss: 0.3339 - val_ce: 0.3339 - va
l_accuracy: 0.8864 - lr: 6.2131e-04
Epoch 90/100
1324/1329 [============================>.] - ETA: 0s - loss: 0.4105 - c
e: 0.4105 - accuracy: 0.8581
Epoch 90: val_accuracy did not improve from 0.90507
1329/1329 [==============================] - 9s 6ms/step - loss: 0.4105
- ce: 0.4105 - accuracy: 0.8580 - val_loss: 0.3146 - val_ce: 0.3146 - va
l_accuracy: 0.8949 - lr: 6.1274e-04
Epoch 91/100
1325/1329 [============================>.] - ETA: 0s - loss: 0.4405 - c
e: 0.4405 - accuracy: 0.8494
Epoch 91: val_accuracy did not improve from 0.90507
1329/1329 [==============================] - 9s 6ms/step - loss: 0.4406
- ce: 0.4406 - accuracy: 0.8493 - val_loss: 0.3363 - val_ce: 0.3363 - va
l_accuracy: 0.8831 - lr: 9.8938e-04
Epoch 92/100
```

```
1329/1329 [==============================] - ETA: 0s - loss: 0.4014 - c
e: 0.4014 - accuracy: 0.8618
Epoch 92: val_accuracy did not improve from 0.90507
1329/1329 [==============================] - 9s 7ms/step - loss: 0.4014
- ce: 0.4014 - accuracy: 0.8618 - val_loss: 0.3067 - val_ce: 0.3067 - va
l_accuracy: 0.8971 - lr: 2.1172e-05
Epoch 93/100
1328/1329 [============================>.] - ETA: 0s - loss: 0.3933 - c
e: 0.3933 - accuracy: 0.8662
Epoch 93: val_accuracy did not improve from 0.90507
1329/1329 [==============================] - 9s 7ms/step - loss: 0.3932
- ce: 0.3932 - accuracy: 0.8662 - val_loss: 0.3049 - val_ce: 0.3049 - va
l_accuracy: 0.9012 - lr: 2.4086e-04
Epoch 94/100
1322/1329 [============================>.] - ETA: 0s - loss: 0.3733 - c
e: 0.3733 - accuracy: 0.8719
Epoch 94: val_accuracy did not improve from 0.90507
1329/1329 [==============================] - 9s 7ms/step - loss: 0.3732
- ce: 0.3732 - accuracy: 0.8719 - val_loss: 0.3011 - val_ce: 0.3011 - va
l_accuracy: 0.8977 - lr: 2.2252e-04
Epoch 95/100
1326/1329 [============================>.] - ETA: 0s - loss: 0.4195 - c
e: 0.4195 - accuracy: 0.8548Restoring model weights from the end of the
best epoch: 85.

Epoch 95: val_accuracy did not improve from 0.90507
1329/1329 [==============================] - 9s 7ms/step - loss: 0.4193
- ce: 0.4193 - accuracy: 0.8548 - val_loss: 0.3995 - val_ce: 0.3995 - va
l_accuracy: 0.8707 - lr: 8.9908e-04
Epoch 95: early stopping

Test Results:
Test Loss: 0.4329
Test Cross-Entropy: 0.4329
Test Accuracy: 0.8624
```




```
🏆 COMPETITION SUMMARY:
📊 Parameters: 118,258 / 122,000 (96.9%)
🎯 Best Validation CE: 0.2847 (Epoch 76)
🎯 Best Validation Accuracy: 0.9051 (Epoch 85)
🎯 Best Validation Loss: 0.2847 (Epoch 76)
📈 Final Test CE: 0.4329
📈 Final Test Accuracy: 0.8624
```

In [3]:
```
import tensorflow as tf
from tensorflow.keras import layers, models, optimizers
import matplotlib.pyplot as plt

# Simple, effective data augmentation
def simple_augmentation(image, label):
```

```
    image = tf.cast(image, tf.float32)
    # Only horizontal flip - keep it simple
    image = tf.image.random_flip_left_right(image)
    return image, label

# Create dataset pipeline
def create_dataset(images, labels, batch_size, is_training=True, validati
    dataset = tf.data.Dataset.from_tensor_slices((images, labels))

    if is_training:
        # Split training data for validation
        dataset_size = len(images)
        val_size = int(dataset_size * validation_split)
        train_size = dataset_size - val_size

        dataset = dataset.shuffle(10000, seed=42)
        train_dataset = dataset.take(train_size)
        val_dataset = dataset.skip(train_size)

        # Apply minimal augmentation only to training data
        train_dataset = train_dataset.cache()
        train_dataset = train_dataset.shuffle(5000, reshuffle_each_iterat
        train_dataset = train_dataset.map(simple_augmentation, num_parall
        train_dataset = train_dataset.batch(batch_size)
        train_dataset = train_dataset.prefetch(tf.data.AUTOTUNE)

        # Validation dataset without augmentation
        val_dataset = val_dataset.batch(batch_size)
        val_dataset = val_dataset.prefetch(tf.data.AUTOTUNE)

        return train_dataset, val_dataset
    else:
        dataset = dataset.batch(batch_size)
        dataset = dataset.prefetch(tf.data.AUTOTUNE)
        return dataset

# Create a parameter-efficient but effective model
def create_simple_effective_model():
    inputs = layers.Input(shape=(32, 32, 3))

    # Block 1 - smaller filters
    x = layers.Conv2D(24, (3, 3), padding='same', activation='relu')(inpu
    x = layers.BatchNormalization()(x)
    x = layers.Conv2D(24, (3, 3), padding='same', activation='relu')(x)
    x = layers.MaxPooling2D((2, 2))(x)
    x = layers.Dropout(0.05)(x)

    # Block 2 - moderate filters
    x = layers.Conv2D(48, (3, 3), padding='same', activation='relu')(x)
    x = layers.BatchNormalization()(x)
    x = layers.Conv2D(48, (3, 3), padding='same', activation='relu')(x)
    x = layers.MaxPooling2D((2, 2))(x)
    x = layers.Dropout(0.05)(x)

    # Block 3 - slightly smaller final block
    x = layers.Conv2D(68, (3, 3), padding='same', activation='relu')(x)
    x = layers.BatchNormalization()(x)
    x = layers.Conv2D(68, (3, 3), padding='same', activation='relu')(x)
    x = layers.Dropout(0.05)(x)
```

```
    # Global average pooling instead of flatten
    x = layers.GlobalAveragePooling2D()(x)

    # Slightly smaller dense layer
    x = layers.Dense(120, activation='relu')(x)  # 128->120
    x = layers.Dropout(0.15)(x)

    outputs = layers.Dense(10)(x)

    model = models.Model(inputs, outputs, name='competitive_model')
    return model

# Simple cosine decay
def cosine_decay_schedule(epoch, total_epochs=100):
    import math
    return 0.001 * 0.5 * (1 + math.cos(math.pi * epoch / total_epochs))

# Build the model
model = create_simple_effective_model()
model.build(input_shape=(None, 32, 32, 3))

# Compile with basic settings
optimizer = optimizers.Adam(learning_rate=0.001)
model.compile(
    optimizer=optimizer,
    loss=tf.keras.losses.SparseCategoricalCrossentropy(from_logits=True),
    metrics=[
        tf.keras.metrics.SparseCategoricalCrossentropy(from_logits=True),
        tf.keras.metrics.SparseCategoricalAccuracy(name='accuracy')
    ]
)

# Display model summary
model.summary()

# Calculate total parameters
total_params = model.count_params()
print(f"Total Parameters: {total_params:,}")
print(f"Parameter budget used: {total_params/122000*100:.1f}%")

# Simple callbacks focused on generalization
callbacks = [
    tf.keras.callbacks.LearningRateScheduler(cosine_decay_schedule),
    tf.keras.callbacks.EarlyStopping(
        monitor='val_accuracy',  # Monitor accuracy instead
        patience=10,
        restore_best_weights=True,
        mode='max',
        verbose=1
    ),
    tf.keras.callbacks.ModelCheckpoint(
        'best_model.h5',
        monitor='val_accuracy',
        save_best_only=True,
        mode='max',
        verbose=1
    )
]

# Load and preprocess CIFAR-10 dataset
```

```
(train_images, train_labels), (test_images, test_labels) = tf.keras.datas

# Normalize pixel values to be between 0 and 1
train_images = train_images.astype('float32') / 255.0
test_images = test_images.astype('float32') / 255.0

# Create datasets with larger validation split
batch_size = 32  # Smaller batch size
train_dataset, val_dataset = create_dataset(train_images, train_labels, b
test_dataset = create_dataset(test_images, test_labels, batch_size, is_tr

# Train the model with fewer epochs
history = model.fit(
    train_dataset,
    epochs=100,  # Much fewer epochs
    validation_data=val_dataset,
    callbacks=callbacks,
    verbose=1
)

# Evaluate on test set
test_results = model.evaluate(test_dataset, verbose=0)
print(f"\nTest Results:")
print(f"Test Loss: {test_results[0]:.4f}")
print(f"Test Cross-Entropy: {test_results[1]:.4f}")
print(f"Test Accuracy: {test_results[2]:.4f}")

# Plot training history with detailed CE information
plt.figure(figsize=(15, 5))

plt.subplot(1, 3, 1)
plt.plot(history.history['ce'], label='Train CE', alpha=0.8)
plt.plot(history.history['val_ce'], label='Val CE', alpha=0.8)

# Find and annotate lowest validation CE
min_val_ce = min(history.history['val_ce'])
min_val_ce_epoch = history.history['val_ce'].index(min_val_ce)
plt.annotate(f'Lowest Val CE: {min_val_ce:.4f}\nEpoch: {min_val_ce_epoch
             xy=(min_val_ce_epoch, min_val_ce),
             xytext=(min_val_ce_epoch + 5, min_val_ce + 0.1),
             arrowprops=dict(arrowstyle='->', color='red', alpha=0.7),
             bbox=dict(boxstyle="round,pad=0.3", facecolor='yellow', alph
             fontsize=10)

# Mark the point
plt.plot(min_val_ce_epoch, min_val_ce, 'ro', markersize=8, alpha=0.8)

plt.title(f'Cross Entropy (Best Val CE: {min_val_ce:.4f})')
plt.xlabel('Epoch')
plt.ylabel('Cross Entropy')
plt.legend()
plt.grid(True, alpha=0.3)

plt.subplot(1, 3, 2)
plt.plot(history.history['accuracy'], label='Train Acc', alpha=0.8)
plt.plot(history.history['val_accuracy'], label='Val Acc', alpha=0.8)

# Find and annotate highest validation accuracy
max_val_acc = max(history.history['val_accuracy'])
max_val_acc_epoch = history.history['val_accuracy'].index(max_val_acc)
```

```
plt.annotate(f'Best Val Acc: {max_val_acc:.4f}\nEpoch: {max_val_acc_epoch
             xy=(max_val_acc_epoch, max_val_acc),
             xytext=(max_val_acc_epoch + 5, max_val_acc - 0.05),
             arrowprops=dict(arrowstyle='->', color='green', alpha=0.7),
             bbox=dict(boxstyle="round,pad=0.3", facecolor='lightgreen',
             fontsize=10)

# Mark the point
plt.plot(max_val_acc_epoch, max_val_acc, 'go', markersize=8, alpha=0.8)

plt.title(f'Accuracy (Best Val Acc: {max_val_acc:.4f})')
plt.xlabel('Epoch')
plt.ylabel('Accuracy')
plt.legend()
plt.grid(True, alpha=0.3)

plt.subplot(1, 3, 3)
plt.plot(history.history['loss'], label='Train Loss', alpha=0.8)
plt.plot(history.history['val_loss'], label='Val Loss', alpha=0.8)

# Find and annotate lowest validation loss
min_val_loss = min(history.history['val_loss'])
min_val_loss_epoch = history.history['val_loss'].index(min_val_loss)
plt.annotate(f'Lowest Val Loss: {min_val_loss:.4f}\nEpoch: {min_val_loss_
             xy=(min_val_loss_epoch, min_val_loss),
             xytext=(min_val_loss_epoch + 5, min_val_loss + 0.1),
             arrowprops=dict(arrowstyle='->', color='blue', alpha=0.7),
             bbox=dict(boxstyle="round,pad=0.3", facecolor='lightblue', a
             fontsize=10)

# Mark the point
plt.plot(min_val_loss_epoch, min_val_loss, 'bo', markersize=8, alpha=0.8)

plt.title(f'Loss (Best Val Loss: {min_val_loss:.4f})')
plt.xlabel('Epoch')
plt.ylabel('Loss')
plt.legend()
plt.grid(True, alpha=0.3)

plt.tight_layout()
plt.show()

# Print summary of best results
print(f"\n🏆 COMPETITION SUMMARY:")
print(f"📊 Parameters: {total_params:,} / 122,000 ({total_params/122000*1
print(f"🎯 Best Validation CE: {min_val_ce:.4f} (Epoch {min_val_ce_epoch
print(f"🎯 Best Validation Accuracy: {max_val_acc:.4f} (Epoch {max_val_ac
print(f"🎯 Best Validation Loss: {min_val_loss:.4f} (Epoch {min_val_loss_
print(f"📈 Final Test CE: {test_results[1]:.4f}")
print(f"📈 Final Test Accuracy: {test_results[2]:.4f}")
```

```
Model: "competitive_model"
_________________________________________________________________
 Layer (type)                Output Shape              Param #   
=================================================================
 input_3 (InputLayer)        [(None, 32, 32, 3)]       0         
                                                                 
 conv2d_6 (Conv2D)           (None, 32, 32, 24)        672       
                                                                 
 batch_normalization_6 (Batc  (None, 32, 32, 24)       96        
 hNormalization)                                                 
                                                                 
 conv2d_7 (Conv2D)           (None, 32, 32, 24)        5208      
                                                                 
 max_pooling2d_4 (MaxPooling  (None, 16, 16, 24)       0         
 2D)                                                             
                                                                 
 dropout_8 (Dropout)         (None, 16, 16, 24)        0         
                                                                 
 conv2d_8 (Conv2D)           (None, 16, 16, 48)        10416     
                                                                 
 batch_normalization_7 (Batc  (None, 16, 16, 48)       192       
 hNormalization)                                                 
                                                                 
 conv2d_9 (Conv2D)           (None, 16, 16, 48)        20784     
                                                                 
 max_pooling2d_5 (MaxPooling  (None, 8, 8, 48)         0         
 2D)                                                             
                                                                 
 dropout_9 (Dropout)         (None, 8, 8, 48)          0         
                                                                 
 conv2d_10 (Conv2D)          (None, 8, 8, 68)          29444     
                                                                 
 batch_normalization_8 (Batc  (None, 8, 8, 68)         272       
 hNormalization)                                                 
                                                                 
 conv2d_11 (Conv2D)          (None, 8, 8, 68)          41684     
                                                                 
 dropout_10 (Dropout)        (None, 8, 8, 68)          0         
                                                                 
 global_average_pooling2d_2   (None, 68)               0         
 (GlobalAveragePooling2D)                                        
                                                                 
 dense_4 (Dense)             (None, 120)               8280      
                                                                 
 dropout_11 (Dropout)        (None, 120)               0         
                                                                 
 dense_5 (Dense)             (None, 10)                1210      
                                                                 
=================================================================
Total params: 118,258
Trainable params: 117,978
Non-trainable params: 280
_________________________________________________________________
Total Parameters: 118,258
Parameter budget used: 96.9%
Epoch 1/100
1325/1329 [============================>.] - ETA: 0s - loss: 1.3419 - c
e: 1.3419 - accuracy: 0.5110
Epoch 1: val_accuracy improved from -inf to 0.60267, saving model to bes
t_model.h5
```

```
1329/1329 [==============================] - 10s 7ms/step - loss: 1.3412
- ce: 1.3412 - accuracy: 0.5112 - val_loss: 1.1086 - val_ce: 1.1086 - va
l_accuracy: 0.6027 - lr: 0.0010
Epoch 2/100
1326/1329 [============================>.] - ETA: 0s - loss: 0.9936 - c
e: 0.9936 - accuracy: 0.6469
Epoch 2: val_accuracy did not improve from 0.60267
1329/1329 [==============================] - 9s 6ms/step - loss: 0.9934
- ce: 0.9934 - accuracy: 0.6470 - val_loss: 1.3650 - val_ce: 1.3650 - va
l_accuracy: 0.5368 - lr: 0.0010
Epoch 3/100
1322/1329 [============================>.] - ETA: 0s - loss: 0.8506 - c
e: 0.8506 - accuracy: 0.7004
Epoch 3: val_accuracy improved from 0.60267 to 0.72013, saving model to
best_model.h5
1329/1329 [==============================] - 9s 7ms/step - loss: 0.8501
- ce: 0.8501 - accuracy: 0.7006 - val_loss: 0.7930 - val_ce: 0.7930 - va
l_accuracy: 0.7201 - lr: 1.0000e-03
Epoch 4/100
1319/1329 [============================>.] - ETA: 0s - loss: 0.7615 - c
e: 0.7615 - accuracy: 0.7355
Epoch 4: val_accuracy did not improve from 0.72013
1329/1329 [==============================] - 9s 6ms/step - loss: 0.7614
- ce: 0.7614 - accuracy: 0.7356 - val_loss: 0.8873 - val_ce: 0.8873 - va
l_accuracy: 0.7016 - lr: 1.0000e-03
Epoch 5/100
1325/1329 [============================>.] - ETA: 0s - loss: 0.6988 - c
e: 0.6988 - accuracy: 0.7589
Epoch 5: val_accuracy did not improve from 0.72013
1329/1329 [==============================] - 9s 7ms/step - loss: 0.6993
- ce: 0.6993 - accuracy: 0.7588 - val_loss: 0.8486 - val_ce: 0.8486 - va
l_accuracy: 0.6973 - lr: 1.0000e-03
Epoch 6/100
1329/1329 [==============================] - ETA: 0s - loss: 0.6497 - c
e: 0.6497 - accuracy: 0.7741
Epoch 6: val_accuracy improved from 0.72013 to 0.77800, saving model to
best_model.h5
1329/1329 [==============================] - 9s 7ms/step - loss: 0.6497
- ce: 0.6497 - accuracy: 0.7741 - val_loss: 0.6549 - val_ce: 0.6549 - va
l_accuracy: 0.7780 - lr: 1.0000e-03
Epoch 7/100
1320/1329 [============================>.] - ETA: 0s - loss: 0.6062 - c
e: 0.6062 - accuracy: 0.7915
Epoch 7: val_accuracy did not improve from 0.77800
1329/1329 [==============================] - 9s 7ms/step - loss: 0.6051
- ce: 0.6051 - accuracy: 0.7918 - val_loss: 0.7846 - val_ce: 0.7846 - va
l_accuracy: 0.7365 - lr: 9.9958e-04
Epoch 8/100
1321/1329 [============================>.] - ETA: 0s - loss: 0.5204 - c
e: 0.5204 - accuracy: 0.8219
Epoch 8: val_accuracy improved from 0.77800 to 0.84147, saving model to
best_model.h5
1329/1329 [==============================] - 9s 7ms/step - loss: 0.5200
- ce: 0.5200 - accuracy: 0.8220 - val_loss: 0.4570 - val_ce: 0.4570 - va
l_accuracy: 0.8415 - lr: 1.8408e-05
Epoch 9/100
1329/1329 [==============================] - ETA: 0s - loss: 0.4900 - c
e: 0.4900 - accuracy: 0.8313
Epoch 9: val_accuracy improved from 0.84147 to 0.85093, saving model to
best_model.h5
```

```
1329/1329 [==============================] - 9s 6ms/step - loss: 0.4900
- ce: 0.4900 - accuracy: 0.8313 - val_loss: 0.4349 - val_ce: 0.4349 - va
l_accuracy: 0.8509 - lr: 7.0725e-05
Epoch 10/100
1324/1329 [============================>.] - ETA: 0s - loss: 0.4772 - c
e: 0.4772 - accuracy: 0.8346
Epoch 10: val_accuracy did not improve from 0.85093
1329/1329 [==============================] - 9s 7ms/step - loss: 0.4770
- ce: 0.4770 - accuracy: 0.8345 - val_loss: 0.4448 - val_ce: 0.4448 - va
l_accuracy: 0.8469 - lr: 2.0896e-04
Epoch 11/100
1328/1329 [============================>.] - ETA: 0s - loss: 0.4570 - c
e: 0.4570 - accuracy: 0.8427
Epoch 11: val_accuracy improved from 0.85093 to 0.85627, saving model to
best_model.h5
1329/1329 [==============================] - 9s 7ms/step - loss: 0.4569
- ce: 0.4569 - accuracy: 0.8427 - val_loss: 0.4196 - val_ce: 0.4196 - va
l_accuracy: 0.8563 - lr: 1.3365e-04
Epoch 12/100
1328/1329 [============================>.] - ETA: 0s - loss: 0.4620 - c
e: 0.4620 - accuracy: 0.8404
Epoch 12: val_accuracy did not improve from 0.85627
1329/1329 [==============================] - 9s 6ms/step - loss: 0.4620
- ce: 0.4620 - accuracy: 0.8404 - val_loss: 0.4684 - val_ce: 0.4684 - va
l_accuracy: 0.8357 - lr: 2.9049e-04
Epoch 13/100
1324/1329 [============================>.] - ETA: 0s - loss: 0.4286 - c
e: 0.4286 - accuracy: 0.8521
Epoch 13: val_accuracy improved from 0.85627 to 0.85840, saving model to
best_model.h5
1329/1329 [==============================] - 9s 7ms/step - loss: 0.4288
- ce: 0.4288 - accuracy: 0.8520 - val_loss: 0.4044 - val_ce: 0.4044 - va
l_accuracy: 0.8584 - lr: 7.9585e-05
Epoch 14/100
1326/1329 [============================>.] - ETA: 0s - loss: 0.5275 - c
e: 0.5275 - accuracy: 0.8183
Epoch 14: val_accuracy did not improve from 0.85840
1329/1329 [==============================] - 9s 6ms/step - loss: 0.5274
- ce: 0.5274 - accuracy: 0.8183 - val_loss: 0.5974 - val_ce: 0.5974 - va
l_accuracy: 0.7912 - lr: 8.7437e-04
Epoch 15/100
1329/1329 [==============================] - ETA: 0s - loss: 0.4646 - c
e: 0.4646 - accuracy: 0.8393
Epoch 15: val_accuracy did not improve from 0.85840
1329/1329 [==============================] - 9s 7ms/step - loss: 0.4646
- ce: 0.4646 - accuracy: 0.8393 - val_loss: 0.4522 - val_ce: 0.4522 - va
l_accuracy: 0.8440 - lr: 5.3975e-04
Epoch 16/100
1323/1329 [============================>.] - ETA: 0s - loss: 0.4627 - c
e: 0.4627 - accuracy: 0.8397
Epoch 16: val_accuracy did not improve from 0.85840
1329/1329 [==============================] - 9s 6ms/step - loss: 0.4625
- ce: 0.4625 - accuracy: 0.8398 - val_loss: 0.4633 - val_ce: 0.4633 - va
l_accuracy: 0.8383 - lr: 6.6456e-04
Epoch 17/100
1327/1329 [============================>.] - ETA: 0s - loss: 0.4947 - c
e: 0.4947 - accuracy: 0.8303
Epoch 17: val_accuracy did not improve from 0.85840
1329/1329 [==============================] - 9s 6ms/step - loss: 0.4946
- ce: 0.4946 - accuracy: 0.8303 - val_loss: 0.4997 - val_ce: 0.4997 - va
```

```
l_accuracy: 0.8307 - lr: 9.9973e-04
Epoch 18/100
1325/1329 [============================>.] - ETA: 0s - loss: 0.3958 - c
e: 0.3958 - accuracy: 0.8609
Epoch 18: val_accuracy improved from 0.85840 to 0.86720, saving model to
best_model.h5
1329/1329 [==============================] - 9s 7ms/step - loss: 0.3955
- ce: 0.3955 - accuracy: 0.8611 - val_loss: 0.3825 - val_ce: 0.3825 - va
l_accuracy: 0.8672 - lr: 2.4941e-04
Epoch 19/100
1324/1329 [============================>.] - ETA: 0s - loss: 0.3768 - c
e: 0.3768 - accuracy: 0.8707
Epoch 19: val_accuracy improved from 0.86720 to 0.87920, saving model to
best_model.h5
1329/1329 [==============================] - 9s 7ms/step - loss: 0.3770
- ce: 0.3770 - accuracy: 0.8707 - val_loss: 0.3627 - val_ce: 0.3627 - va
l_accuracy: 0.8792 - lr: 2.7019e-04
Epoch 20/100
1324/1329 [============================>.] - ETA: 0s - loss: 0.4224 - c
e: 0.4224 - accuracy: 0.8546
Epoch 20: val_accuracy did not improve from 0.87920
1329/1329 [==============================] - 9s 6ms/step - loss: 0.4226
- ce: 0.4226 - accuracy: 0.8544 - val_loss: 0.5117 - val_ce: 0.5117 - va
l_accuracy: 0.8261 - lr: 6.9197e-04
Epoch 21/100
1321/1329 [============================>.] - ETA: 0s - loss: 0.3729 - c
e: 0.3729 - accuracy: 0.8721
Epoch 21: val_accuracy improved from 0.87920 to 0.88253, saving model to
best_model.h5
1329/1329 [==============================] - 9s 7ms/step - loss: 0.3727
- ce: 0.3727 - accuracy: 0.8721 - val_loss: 0.3522 - val_ce: 0.3522 - va
l_accuracy: 0.8825 - lr: 1.3757e-05
Epoch 22/100
1327/1329 [============================>.] - ETA: 0s - loss: 0.4565 - c
e: 0.4565 - accuracy: 0.8416
Epoch 22: val_accuracy did not improve from 0.88253
1329/1329 [==============================] - 9s 7ms/step - loss: 0.4565
- ce: 0.4565 - accuracy: 0.8416 - val_loss: 0.4655 - val_ce: 0.4655 - va
l_accuracy: 0.8439 - lr: 9.9646e-04
Epoch 23/100
1325/1329 [============================>.] - ETA: 0s - loss: 0.4269 - c
e: 0.4269 - accuracy: 0.8505
Epoch 23: val_accuracy did not improve from 0.88253
1329/1329 [==============================] - 9s 7ms/step - loss: 0.4270
- ce: 0.4270 - accuracy: 0.8506 - val_loss: 0.4622 - val_ce: 0.4622 - va
l_accuracy: 0.8457 - lr: 9.0587e-04
Epoch 24/100
1327/1329 [============================>.] - ETA: 0s - loss: 0.4229 - c
e: 0.4229 - accuracy: 0.8529
Epoch 24: val_accuracy did not improve from 0.88253
1329/1329 [==============================] - 9s 6ms/step - loss: 0.4233
- ce: 0.4233 - accuracy: 0.8528 - val_loss: 0.4292 - val_ce: 0.4292 - va
l_accuracy: 0.8569 - lr: 9.9776e-04
Epoch 25/100
1328/1329 [============================>.] - ETA: 0s - loss: 0.4202 - c
e: 0.4202 - accuracy: 0.8552
Epoch 25: val_accuracy did not improve from 0.88253
1329/1329 [==============================] - 9s 6ms/step - loss: 0.4202
- ce: 0.4202 - accuracy: 0.8552 - val_loss: 0.4173 - val_ce: 0.4173 - va
l_accuracy: 0.8571 - lr: 9.9978e-04
```

```
Epoch 26/100
1324/1329 [============================>.] - ETA: 0s - loss: 0.3523 - c
e: 0.3523 - accuracy: 0.8775
Epoch 26: val_accuracy did not improve from 0.88253
1329/1329 [==============================] - 9s 7ms/step - loss: 0.3525
- ce: 0.3525 - accuracy: 0.8773 - val_loss: 0.3784 - val_ce: 0.3784 - va
l_accuracy: 0.8731 - lr: 5.3757e-04
Epoch 27/100
1329/1329 [==============================] - ETA: 0s - loss: 0.3883 - c
e: 0.3883 - accuracy: 0.8664
Epoch 27: val_accuracy did not improve from 0.88253
1329/1329 [==============================] - 9s 6ms/step - loss: 0.3883
- ce: 0.3883 - accuracy: 0.8664 - val_loss: 0.3999 - val_ce: 0.3999 - va
l_accuracy: 0.8693 - lr: 9.5489e-04
Epoch 28/100
1327/1329 [============================>.] - ETA: 0s - loss: 0.3154 - c
e: 0.3154 - accuracy: 0.8910
Epoch 28: val_accuracy improved from 0.88253 to 0.89347, saving model to
best_model.h5
1329/1329 [==============================] - 9s 7ms/step - loss: 0.3155
- ce: 0.3155 - accuracy: 0.8910 - val_loss: 0.3269 - val_ce: 0.3269 - va
l_accuracy: 0.8935 - lr: 3.4819e-04
Epoch 29/100
1325/1329 [============================>.] - ETA: 0s - loss: 0.2803 - c
e: 0.2803 - accuracy: 0.9039
Epoch 29: val_accuracy improved from 0.89347 to 0.90293, saving model to
best_model.h5
1329/1329 [==============================] - 9s 6ms/step - loss: 0.2800
- ce: 0.2800 - accuracy: 0.9040 - val_loss: 0.2941 - val_ce: 0.2941 - va
l_accuracy: 0.9029 - lr: 1.6462e-04
Epoch 30/100
1326/1329 [============================>.] - ETA: 0s - loss: 0.3011 - c
e: 0.3011 - accuracy: 0.8952
Epoch 30: val_accuracy did not improve from 0.90293
1329/1329 [==============================] - 9s 6ms/step - loss: 0.3011
- ce: 0.3011 - accuracy: 0.8952 - val_loss: 0.3546 - val_ce: 0.3546 - va
l_accuracy: 0.8815 - lr: 4.6585e-04
Epoch 31/100
1321/1329 [============================>.] - ETA: 0s - loss: 0.3594 - c
e: 0.3594 - accuracy: 0.8748
Epoch 31: val_accuracy did not improve from 0.90293
1329/1329 [==============================] - 8s 6ms/step - loss: 0.3596
- ce: 0.3596 - accuracy: 0.8746 - val_loss: 0.3629 - val_ce: 0.3629 - va
l_accuracy: 0.8779 - lr: 9.4167e-04
Epoch 32/100
1325/1329 [============================>.] - ETA: 0s - loss: 0.3356 - c
e: 0.3356 - accuracy: 0.8822
Epoch 32: val_accuracy did not improve from 0.90293
1329/1329 [==============================] - 9s 7ms/step - loss: 0.3361
- ce: 0.3361 - accuracy: 0.8820 - val_loss: 0.4485 - val_ce: 0.4485 - va
l_accuracy: 0.8455 - lr: 7.6562e-04
Epoch 33/100
1328/1329 [============================>.] - ETA: 0s - loss: 0.3526 - c
e: 0.3526 - accuracy: 0.8781
Epoch 33: val_accuracy did not improve from 0.90293
1329/1329 [==============================] - 9s 6ms/step - loss: 0.3526
- ce: 0.3526 - accuracy: 0.8781 - val_loss: 0.3515 - val_ce: 0.3515 - va
l_accuracy: 0.8811 - lr: 9.6324e-04
Epoch 34/100
1325/1329 [============================>.] - ETA: 0s - loss: 0.3024 - c
```

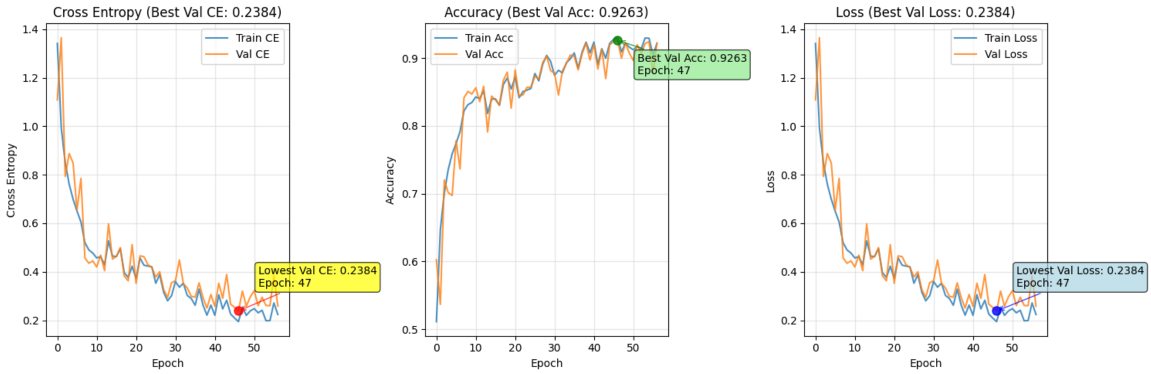
```
e: 0.3024 - accuracy: 0.8929
Epoch 34: val_accuracy did not improve from 0.90293
1329/1329 [==============================] - 9s 6ms/step - loss: 0.3022
- ce: 0.3022 - accuracy: 0.8931 - val_loss: 0.3333 - val_ce: 0.3333 - va
l_accuracy: 0.8919 - lr: 5.7653e-04
Epoch 35/100
1320/1329 [============================>.] - ETA: 0s - loss: 0.2901 - c
e: 0.2901 - accuracy: 0.8989
Epoch 35: val_accuracy improved from 0.90293 to 0.90453, saving model to
best_model.h5
1329/1329 [==============================] - 9s 7ms/step - loss: 0.2900
- ce: 0.2900 - accuracy: 0.8990 - val_loss: 0.2989 - val_ce: 0.2989 - va
l_accuracy: 0.9045 - lr: 5.5901e-04
Epoch 36/100
1321/1329 [============================>.] - ETA: 0s - loss: 0.2621 - c
e: 0.2621 - accuracy: 0.9076
Epoch 36: val_accuracy did not improve from 0.90453
1329/1329 [==============================] - 9s 7ms/step - loss: 0.2617
- ce: 0.2617 - accuracy: 0.9077 - val_loss: 0.2950 - val_ce: 0.2950 - va
l_accuracy: 0.9027 - lr: 3.8513e-04
Epoch 37/100
1321/1329 [============================>.] - ETA: 0s - loss: 0.3281 - c
e: 0.3281 - accuracy: 0.8852
Epoch 37: val_accuracy did not improve from 0.90453
1329/1329 [==============================] - 9s 7ms/step - loss: 0.3284
- ce: 0.3284 - accuracy: 0.8851 - val_loss: 0.3542 - val_ce: 0.3542 - va
l_accuracy: 0.8825 - lr: 9.9939e-04
Epoch 38/100
1324/1329 [============================>.] - ETA: 0s - loss: 0.2626 - c
e: 0.2626 - accuracy: 0.9088
Epoch 38: val_accuracy improved from 0.90453 to 0.90640, saving model to
best_model.h5
1329/1329 [==============================] - 9s 7ms/step - loss: 0.2623
- ce: 0.2623 - accuracy: 0.9089 - val_loss: 0.2935 - val_ce: 0.2935 - va
l_accuracy: 0.9064 - lr: 3.5214e-04
Epoch 39/100
1322/1329 [============================>.] - ETA: 0s - loss: 0.2212 - c
e: 0.2212 - accuracy: 0.9234
Epoch 39: val_accuracy improved from 0.90640 to 0.92227, saving model to
best_model.h5
1329/1329 [==============================] - 9s 7ms/step - loss: 0.2209
- ce: 0.2209 - accuracy: 0.9236 - val_loss: 0.2502 - val_ce: 0.2502 - va
l_accuracy: 0.9223 - lr: 8.8288e-05
Epoch 40/100
1325/1329 [============================>.] - ETA: 0s - loss: 0.2623 - c
e: 0.2623 - accuracy: 0.9077
Epoch 40: val_accuracy did not improve from 0.92227
1329/1329 [==============================] - 9s 7ms/step - loss: 0.2624
- ce: 0.2624 - accuracy: 0.9076 - val_loss: 0.3063 - val_ce: 0.3063 - va
l_accuracy: 0.8972 - lr: 5.3810e-04
Epoch 41/100
1321/1329 [============================>.] - ETA: 0s - loss: 0.2206 - c
e: 0.2206 - accuracy: 0.9236
Epoch 41: val_accuracy did not improve from 0.92227
1329/1329 [==============================] - 9s 7ms/step - loss: 0.2203
- ce: 0.2203 - accuracy: 0.9237 - val_loss: 0.2570 - val_ce: 0.2570 - va
l_accuracy: 0.9197 - lr: 7.1154e-05
Epoch 42/100
1320/1329 [============================>.] - ETA: 0s - loss: 0.3051 - c
e: 0.3051 - accuracy: 0.8908
```

```
Epoch 42: val_accuracy did not improve from 0.92227
1329/1329 [==============================] - 9s 6ms/step - loss: 0.3052
- ce: 0.3052 - accuracy: 0.8908 - val_loss: 0.3524 - val_ce: 0.3524 - va
l_accuracy: 0.8839 - lr: 9.0997e-04
Epoch 43/100
1325/1329 [============================>.] - ETA: 0s - loss: 0.2472 - c
e: 0.2472 - accuracy: 0.9137
Epoch 43: val_accuracy did not improve from 0.92227
1329/1329 [==============================] - 9s 6ms/step - loss: 0.2468
- ce: 0.2468 - accuracy: 0.9138 - val_loss: 0.2907 - val_ce: 0.2907 - va
l_accuracy: 0.9131 - lr: 3.0913e-04
Epoch 44/100
1323/1329 [============================>.] - ETA: 0s - loss: 0.2824 - c
e: 0.2824 - accuracy: 0.8998
Epoch 44: val_accuracy did not improve from 0.92227
1329/1329 [==============================] - 9s 7ms/step - loss: 0.2823
- ce: 0.2823 - accuracy: 0.8998 - val_loss: 0.3882 - val_ce: 0.3882 - va
l_accuracy: 0.8697 - lr: 8.0221e-04
Epoch 45/100
1320/1329 [============================>.] - ETA: 0s - loss: 0.2258 - c
e: 0.2258 - accuracy: 0.9217
Epoch 45: val_accuracy did not improve from 0.92227
1329/1329 [==============================] - 9s 7ms/step - loss: 0.2257
- ce: 0.2257 - accuracy: 0.9217 - val_loss: 0.2667 - val_ce: 0.2667 - va
l_accuracy: 0.9193 - lr: 2.0688e-04
Epoch 46/100
1328/1329 [============================>.] - ETA: 0s - loss: 0.2092 - c
e: 0.2092 - accuracy: 0.9277
Epoch 46: val_accuracy improved from 0.92227 to 0.92440, saving model to
best_model.h5
1329/1329 [==============================] - 9s 7ms/step - loss: 0.2092
- ce: 0.2092 - accuracy: 0.9277 - val_loss: 0.2530 - val_ce: 0.2530 - va
l_accuracy: 0.9244 - lr: 2.3108e-04
Epoch 47/100
1324/1329 [============================>.] - ETA: 0s - loss: 0.1942 - c
e: 0.1942 - accuracy: 0.9301
Epoch 47: val_accuracy improved from 0.92440 to 0.92627, saving model to
best_model.h5
1329/1329 [==============================] - 9s 7ms/step - loss: 0.1942
- ce: 0.1942 - accuracy: 0.9301 - val_loss: 0.2384 - val_ce: 0.2384 - va
l_accuracy: 0.9263 - lr: 6.8688e-06
Epoch 48/100
1329/1329 [==============================] - ETA: 0s - loss: 0.2543 - c
e: 0.2543 - accuracy: 0.9090
Epoch 48: val_accuracy did not improve from 0.92627
1329/1329 [==============================] - 9s 7ms/step - loss: 0.2543
- ce: 0.2543 - accuracy: 0.9090 - val_loss: 0.3203 - val_ce: 0.3203 - va
l_accuracy: 0.8999 - lr: 6.4382e-04
Epoch 49/100
1326/1329 [============================>.] - ETA: 0s - loss: 0.2203 - c
e: 0.2203 - accuracy: 0.9220
Epoch 49: val_accuracy did not improve from 0.92627
1329/1329 [==============================] - 9s 7ms/step - loss: 0.2202
- ce: 0.2202 - accuracy: 0.9220 - val_loss: 0.2578 - val_ce: 0.2578 - va
l_accuracy: 0.9223 - lr: 7.8052e-06
Epoch 50/100
1320/1329 [============================>.] - ETA: 0s - loss: 0.2384 - c
e: 0.2384 - accuracy: 0.9161
Epoch 50: val_accuracy did not improve from 0.92627
1329/1329 [==============================] - 9s 7ms/step - loss: 0.2385
```

```
- ce: 0.2385 - accuracy: 0.9161 - val_loss: 0.2966 - val_ce: 0.2966 - va
l_accuracy: 0.9072 - lr: 5.3536e-04
Epoch 51/100
1324/1329 [============================>.] - ETA: 0s - loss: 0.2482 - c
e: 0.2482 - accuracy: 0.9122
Epoch 51: val_accuracy did not improve from 0.92627
1329/1329 [==============================] - 9s 6ms/step - loss: 0.2480
- ce: 0.2480 - accuracy: 0.9122 - val_loss: 0.3219 - val_ce: 0.3219 - va
l_accuracy: 0.8964 - lr: 6.7547e-04
Epoch 52/100
1329/1329 [==============================] - ETA: 0s - loss: 0.2307 - c
e: 0.2307 - accuracy: 0.9174
Epoch 52: val_accuracy did not improve from 0.92627
1329/1329 [==============================] - 9s 6ms/step - loss: 0.2307
- ce: 0.2307 - accuracy: 0.9174 - val_loss: 0.2603 - val_ce: 0.2603 - va
l_accuracy: 0.9197 - lr: 5.1833e-04
Epoch 53/100
1321/1329 [============================>.] - ETA: 0s - loss: 0.2411 - c
e: 0.2411 - accuracy: 0.9136
Epoch 53: val_accuracy did not improve from 0.92627
1329/1329 [==============================] - 9s 6ms/step - loss: 0.2412
- ce: 0.2412 - accuracy: 0.9135 - val_loss: 0.2949 - val_ce: 0.2949 - va
l_accuracy: 0.9068 - lr: 6.8508e-04
Epoch 54/100
1320/1329 [============================>.] - ETA: 0s - loss: 0.1975 - c
e: 0.1975 - accuracy: 0.9298
Epoch 54: val_accuracy did not improve from 0.92627
1329/1329 [==============================] - 9s 6ms/step - loss: 0.1978
- ce: 0.1978 - accuracy: 0.9297 - val_loss: 0.2611 - val_ce: 0.2611 - va
l_accuracy: 0.9213 - lr: 2.2975e-04
Epoch 55/100
1325/1329 [============================>.] - ETA: 0s - loss: 0.1984 - c
e: 0.1984 - accuracy: 0.9294
Epoch 55: val_accuracy did not improve from 0.92627
1329/1329 [==============================] - 9s 7ms/step - loss: 0.1987
- ce: 0.1987 - accuracy: 0.9293 - val_loss: 0.2598 - val_ce: 0.2598 - va
l_accuracy: 0.9244 - lr: 3.7783e-04
Epoch 56/100
1324/1329 [============================>.] - ETA: 0s - loss: 0.2713 - c
e: 0.2713 - accuracy: 0.9043
Epoch 56: val_accuracy did not improve from 0.92627
1329/1329 [==============================] - 9s 7ms/step - loss: 0.2714
- ce: 0.2714 - accuracy: 0.9043 - val_loss: 0.3830 - val_ce: 0.3830 - va
l_accuracy: 0.8776 - lr: 9.9988e-04
Epoch 57/100
1325/1329 [============================>.] - ETA: 0s - loss: 0.2241 - c
e: 0.2241 - accuracy: 0.9214Restoring model weights from the end of the
best epoch: 47.

Epoch 57: val_accuracy did not improve from 0.92627
1329/1329 [==============================] - 9s 7ms/step - loss: 0.2240
- ce: 0.2240 - accuracy: 0.9214 - val_loss: 0.2585 - val_ce: 0.2585 - va
l_accuracy: 0.9223 - lr: 2.3470e-05
Epoch 57: early stopping

Test Results:
Test Loss: 0.4933
Test Cross-Entropy: 0.4933
Test Accuracy: 0.8543
```



🏆 COMPETITION SUMMARY:
📊 Parameters: 118,258 / 122,000 (96.9%)
🎯 Best Validation CE: 0.2384 (Epoch 47)
🎯 Best Validation Accuracy: 0.9263 (Epoch 47)
🎯 Best Validation Loss: 0.2384 (Epoch 47)
📈 Final Test CE: 0.4933
📈 Final Test Accuracy: 0.8543

In [ ]: