Classifying Alzheimer disease from MRI scans using DL

CSCA 5642: Introduction to Deep Learning

PHPO 4876: Philipp Adrian Pohlmann

University of Colorado **Boulder**

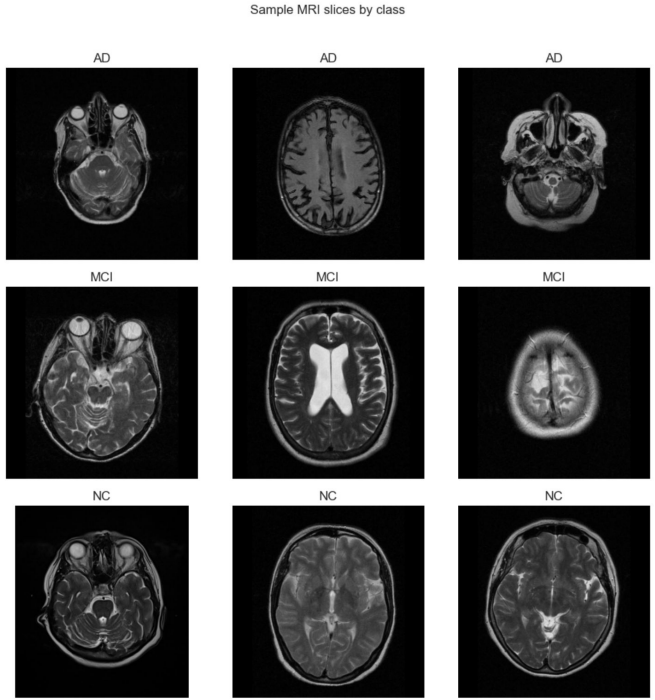
The Problem: Early Alzheimer's Disease Detection

The Challenge:

- Alzheimer's affects millions, early detection is critical
- Manual MRI analysis is time-intensive and requires expertise
- Subtle anatomical changes are difficult to detect visually

Deep Learning Approach:

- Classify brain MRI slices: AD, MCI, and Normal Control
- Compare CNNs from scratch vs transfer learning
- Understand model complexity vs dataset size trade-offs



The Dataset: DICOM MRI Scans

Dataset Characteristics

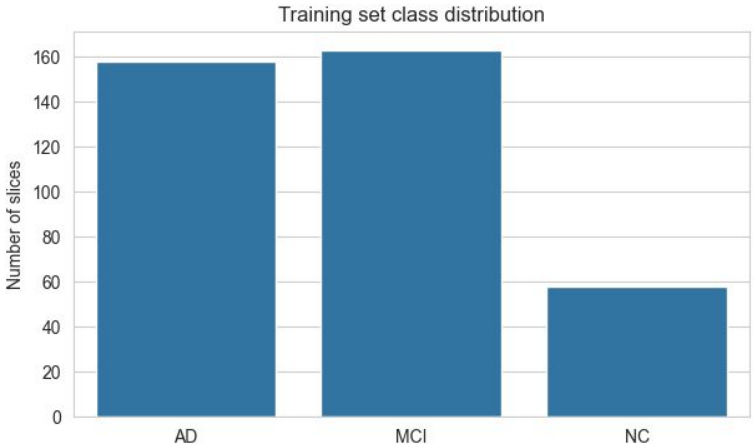
Total Size: 474 samples

- AD: 197 samples, MCI: 204 samples, NC: 73 samples
- Class Imbalance: NC underrepresented (common in clinical data)

Data Split:

- 329 training, 67 validation, 48 test samples

Preprocessing: Normalized to [0,1], resized to 160×160×3



Exploratory Data Analysis

EDA Findings

- ➔ No missing data or corrupted DICOM files
- ➔ Sample MRI slices show varying brain regions and anatomical features
- ➔ Class imbalance noted: NC (73) vs AD/MCI (~200 each)
- ➔ Subtle differences visible across diagnostic categories

Preprocessing Decisions:

- Normalize to [0,1] for stable gradient descent
- Resize to 160×160 to preserve anatomical detail
- Use macro F1 score to handle class imbalance

```python
def load_dicom_for_display(path):
    ds = pydicom.dcmread(path)
    img = ds.pixel_array.astype(np.float32)
    img = img - np.min(img)
    if np.max(img) > 0:
        img = img / np.max(img)
    return img

def show_examples_per_class(file_paths, labels, class_names, num_per_class=3):
    plt.figure(figsize=(num_per_class * 3, len(class_names) * 3))
    idx = 1
    for class_id, class_name in enumerate[Any](class_names):
        class_indices = np.where(labels == class_id)[0]
        np.random.shuffle(class_indices)
        for i in range(num_per_class):
            if i >= len(class_indices):
                continue
            path = file_paths[class_indices[i]]
            img = load_dicom_for_display(path)
            ax = plt.subplot(len(class_names), num_per_class, idx)
            plt.imshow(img, cmap="gray")
            plt.title(class_name)
            plt.axis("off")
            idx += 1
    plt.suptitle("Sample MRI slices by class", y=1.02, fontsize=12)
    plt.tight_layout()
    plt.show()

show_examples_per_class(train_paths, train_labels, class_names, num_per_class=3)
```

Python

Three CNN Models Tested

Model Comparison Strategy

1. Baseline CNN (~5M parameters):

- Standard architecture: Conv2D + Max Pooling + Dropout
- Low capacity to prevent overfitting on small dataset

2. Deep CNN (~6.5M parameters):

- Deeper with batch normalization and dropout
- Tests if additional capacity helps capture MRI features

3. Transfer Learning (MobileNetV2):

- ImageNet pretrained weights, frozen then fine-tuned
- Tests domain transfer from natural to medical images

	Model	Val accuracy	Val macro F1
0	Baseline CNN	0.957447	0.965789
1	Deep CNN	0.510638	0.373016
2	Transfer learning	0.468085	0.340000

Results: Baseline CNN Performs Best

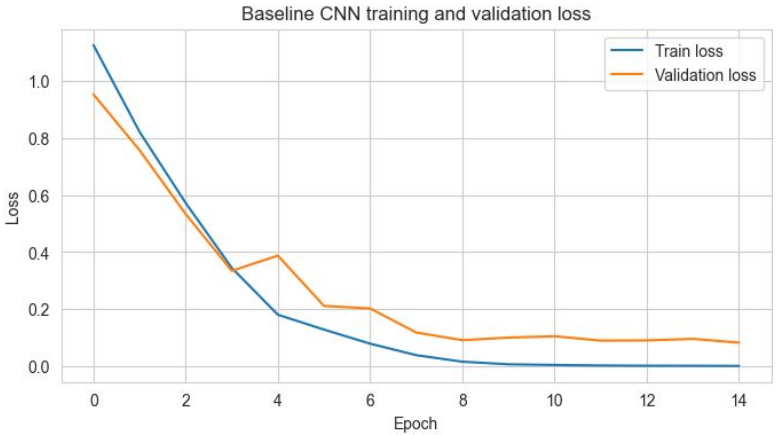
Validation Performance

Winner:

- Baseline CNN: Val Acc = 95.7% | Macro F1 = 96.6%

Underperformers:

- Deep CNN: Val Acc = 51.1% (severe overfitting despite batch norm)
- Transfer Learning: Val Acc = 46.8% (domain mismatch)



Key Finding: Simple model with 5M params beat complex 6.5M param model

Test Set Performance: Strong Generalization

Baseline CNN on Test Set

Final Performance:

- ➔ Test Accuracy: 85.4% on 48 held-out samples
- ➔ Test Macro F1: 0.855 (balanced across all classes)
- ➔ Validation-Test gap: 10.3% (95.7% → 85.4%)
- ➔ Training curves show stable convergence with early stopping

Significance:

- 85% accuracy suitable for pre-screening tool
- Could flag cases requiring expert radiologist review

```python
# Evaluate the best model (Baseline CNN) on the test set

y_true_test = []
y_pred_test = []

for images, labels in test_ds:
    preds = baseline_model.predict(images, verbose=0)
    y_true_test.extend(labels.numpy())
    y_pred_test.extend(np.argmax(preds, axis=1))

y_true_test = np.array(y_true_test)
y_pred_test = np.array(y_pred_test)

test_acc = accuracy_score(y_true_test, y_pred_test)
test_f1 = f1_score(y_true_test, y_pred_test, average="macro")

print("Baseline CNN – test set performance")
print(f"Test accuracy: {test_acc:.3f}")
print(f"Macro F1:      {test_f1:.3f}")
```

Python

```
Baseline CNN – test set performance
Test accuracy: 0.854
Macro F1:      0.855
```

Analysis: Why did the Baseline Model win?

Understanding Model Performance

1. Baseline CNN matched dataset size:

- 474 samples → 5M parameters is appropriate capacity
- Task-specific design learns relevant spatial features

2. Deep CNN suffered from excess capacity:

- 6.5M params too many → memorized training data
- Batch normalization couldn't overcome fundamental overfitting

3. Transfer learning hit domain mismatch:

- ImageNet features don't transfer to grayscale MRI anatomy
- Fine-tuning helped but couldn't overcome initial mismatch

Layer (type)	Output Shape	Param #
conv2d_8 (Conv2D)	(None, 158, 158, 32)	896
max_pooling2d_8 (MaxPooling2D)	(None, 79, 79, 32)	0
conv2d_9 (Conv2D)	(None, 77, 77, 64)	18,496
max_pooling2d_9 (MaxPooling2D)	(None, 38, 38, 64)	0
flatten_3 (Flatten)	(None, 92416)	0
dense_8 (Dense)	(None, 64)	5,914,688
dense_9 (Dense)	(None, 3)	195

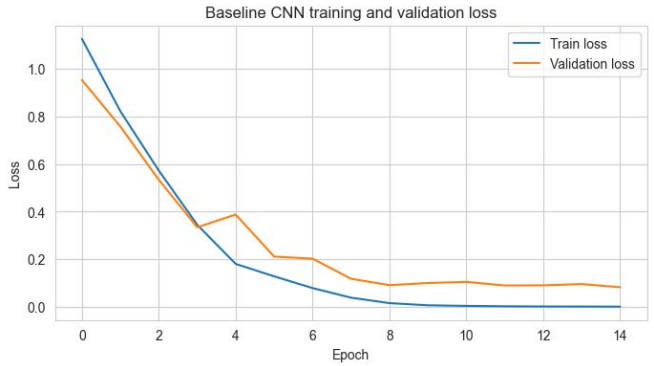
Training Configuration

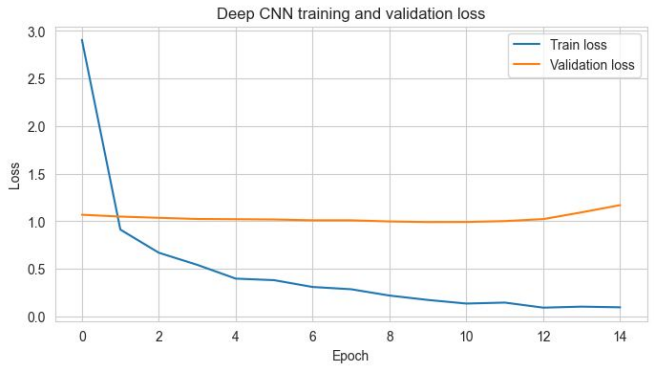
Training Details

- Optimizer: Adam with default learning rate
- Loss: Categorical cross-entropy
- Early stopping: Patience = 10 epochs on validation loss
- Data augmentation: Random flips, rotations, zoom

Regularization:

- Dropout (0.5) after pooling and dense layers
- Batch normalization in Deep CNN (still overfitted)
- L2 regularization tested but not in final models





Key Takeaways

#1 Model complexity must match dataset size

- Baseline (5M params): 95.7% val, 85.4% test accuracy
- Deep CNN (6.5M params): Only 51% validation accuracy
- 474 samples → compact models prevent overfitting

#2 Domain-specific training beats transfer learning

- MobileNetV2 (ImageNet): Only 46.8% accuracy
- Natural image features don't transfer to medical imaging

#3 CNNs can learn meaningful patterns from MRI

- 85% test accuracy demonstrates real-world potential
- Spatial features matter more than raw intensity values

Thank You!

University of Colorado **Boulder**