Print PDF

Page 1: Introduction to CNN and RCNN

# Introduction to CNN and RCNN

## What is CNN (Convolutional Neural Network)?

CNNs are a class of deep neural networks widely used in image classification, object detection, and video analysis. They are biologically inspired by the visual cortex and can automatically detect important features in an image, such as edges, colors, and patterns.

### Basic Layers of CNN:

- **Convolution Layer** – Applies filters to extract features like edges and textures.
- **ReLU (Activation)** – Adds non-linearity.
- **Pooling Layer** – Reduces spatial dimensions and computation.
- **Fully Connected Layer** – Performs classification based on extracted features.

### Use Cases:

- Image classification (e.g., cat vs dog)
- Face recognition
- Medical image analysis

## What is RCNN (Region-based CNN)?

RCNN is an extension of CNN, designed specifically for object detection — identifying what objects are in an image and where they are located. It combines region proposal algorithms with CNNs for localizing multiple objects in an image.

### Key Idea:

> Instead of feeding the entire image directly into a CNN, RCNN first proposes candidate regions (bounding boxes) where objects might be, then uses a CNN to classify each one.

**Use Cases:**

- Object detection in autonomous driving
- Surveillance
- Satellite imagery analysis (like your cyclone intensity project)





# CNN Architecture and Workflow

## 1. Convolutional Layer

Performs element-wise multiplication between a filter (kernel) and a patch of the input image.

Extracts low-level features.

```
Example:
Input: 5x5 image → Filter: 3x3 → Output: 3x3 feature map
```

## 2. Activation Function (ReLU)

- Applies **f(x) = max(0, x)**
- Adds non-linearity to model complex patterns.

## 3. Pooling Layer

- **Types:** Max Pooling, Average Pooling
- Reduces dimensionality while preserving important features.

### 4. Fully Connected Layer

- Takes all neurons from the previous layer and connects them to each neuron in the next layer.
- Outputs a probability score for each class (via softmax).

### 5. Training with Backpropagation

- CNN uses gradient descent to minimize the loss.
- Weights in convolution layers are updated using backpropagation.

**Advantages of CNN:**

- Automatic feature extraction
- Translational invariance
- Requires fewer parameters than fully connected networks





## RCNN Architecture

### Why Not Just CNN for Object Detection?

- CNNs can classify **what** is in an image but not **where**.
- RCNN solves this by adding region proposals.

## RCNN Pipeline:

### 1. Region Proposal using Selective Search

Finds ~2000 region proposals likely to contain objects.

### 2. Feature Extraction using CNN

Each region is resized and passed through a CNN.

### 3. Classification using SVM

RCNN originally used a Support Vector Machine (SVM) to classify each region.

### 4. Bounding Box Regression

Fine-tunes box coordinates for better accuracy.

## Training:

- Multi-stage process (CNN, SVM, bounding box regressor trained separately).
- High accuracy but very slow due to separate passes for each region.

**Key Terms:**

- **IoU (Intersection over Union):** Measures overlap between predicted and ground truth boxes.
- **Region Proposals:** Candidate image regions likely to contain an object.

Page 3

Page 4: Variants and Improvements

# Variants and Improvements

## Fast RCNN

- Solves RCNN's inefficiency.
- Entire image passed through CNN once; region proposals extracted from shared feature map.
- Classification and bounding box regression done together.

## Faster RCNN

- Introduces Region Proposal Network (RPN) — a small CNN that proposes regions.
- Entire process becomes end-to-end and faster.

## Mask RCNN (for segmentation)

Adds a third branch for predicting object masks (segmentation) along with classification and bounding box.

## Differences: CNN vs RCNN

| Feature | CNN | RCNN |
|---|---|---|
| **Task** | Classification | Detection |
| **Input** | Entire image | Region proposals |
| **Speed** | Fast | Slow (RCNN) |
| **Complexity** | Simple | Multi-stage |
| **Output** | Class label | Class + Bounding box |

Page 5: Applications, Limitations, and Conclusion

# Applications, Limitations, and Conclusion

## Applications of CNN

- Medical diagnosis (tumor detection)
- OCR (Optical Character Recognition)
- Style transfer, super-resolution
- Face verification

## Applications of RCNN

- Self-driving cars (pedestrian and vehicle detection)
- Security cameras (intrusion detection)
- Satellite and drone imagery (damage assessment, cyclone tracking)

## Limitations:

### CNN:

- Cannot handle object location or multiple objects well
- Sensitive to rotation and scale unless augmented

### RCNN:

- Original RCNN is computationally expensive
- Needs large dataset and long training times
- Not suitable for real-time applications unless using Fast/Faster RCNN

## Conclusion:

CNN is a foundational deep learning model that is excellent at image classification and feature extraction. RCNN builds on this by adding region proposals, making it powerful for object detection tasks. While RCNN introduced high accuracy in detecting multiple objects, its successors like Fast RCNN and Faster RCNN made it practical for real-world use by making the system faster and end-to-end trainable.