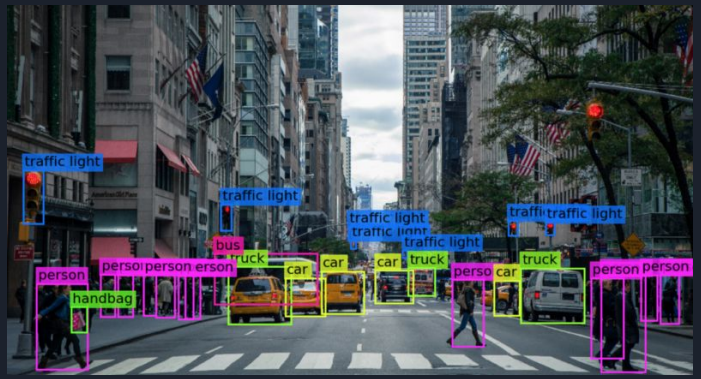
Computer Vision Presentation

Brad Kim, Jimmy Nguyen, Ryan Nguyen

What is CV?

Computer Vision is a field of artificial intelligence (AI) that enables computers to derive meaningful information from **digital images**, **videos**, and other **visual inputs**



How does it work?

Computer vision applications run on algorithms that use massive amounts of visual data or images. They analyze the data given to them and try to recognize patterns to determine the content of other images.

General Steps to CV:

I. **Image Acquisition** - gathering visual data through cameras or sensors
II. **Image Processing** - analyzing visual data to identify patterns or features
III. **Object Recognition** - categorizing and identifying objects in images/videos

Common Use Cases:

- Facial Recognition
- Object Detection & Recognition
- Industrial Quality Control
- Augmented Reality (AR)
- Sports Analytics
- Security & Surveillance

Implementation?

“Two essential technologies are used to accomplish [computer vision]: a type of machine learning called **deep learning** and a **convolutional neural network (CNN).**”

How It Applies To Inkwave?

- Computer Vision allows our program to extract machine-readable text from image
- Use input device like a camera to obtain data
- Specific field within computer vision, Optical Character Recognition (OCR), for extracting text from images
  - OCR algorithms can locate and draw boxes on image where text is to then turn into machine-readable text
- Not only “sees” but also can output text translated from image

- OCR systems uses machine learning models to train large data sets and interpret extracted features
  - Recurrent Neural Network (RNN) - time series, speech, text, and financial data
  - Pros:
    - Uses sequential memory to take input and compare with past data to identify text
    - If handwriting is different, it is better at identifying than other models like CNN or DNN
  - Con:
    - Vanishing Gradient Problem
      - RNN learns using backpropagation algorithm and gradients learned during training become small and barely updates the model, which creates a slow-learning model since RNN has many layers and backpropagation is exponentially so gradients gets exponentially smaller
- Solution:
  - Long Short-Term Memory (LSTM)
  - Pro:
    - Type of RNN algorithm that is designed to not have Vanishing Gradient Problem
    - Has gating mechanism that controls flow of information so that gradient doesn’t become too small that it doesn’t barely update model
    - Has Memory Cells so it can store and access information over long periods
  - Con:
    - It is harder to implement and understand than the simple RNN since much more complex
    - Also takes more computational power and time to interpret data

OCR Pipeline

- Preprocessing
  - Cleaning dataset
- Text recognition
  - Pattern matching - Ex: Isolate characters w/ glyphs
  - Feature extraction - Ex: Decomposes glyphs, finds best match
- Post processing
  - Extracted text data -> Computerized File

OCR Types

- Simple OCR software (char by char)
- Intelligent character recognition software (Simple OCR Improved)
- Intelligent word recognition (Process whole word images)
- Optical mark recognition (Recognizes logos, watermarks, etc)

https://nanonets.com/blog/attention-ocr-for-text-recogntion/

https://medium.com/analytics-vidhya/optical-character-recognition-using-tensorflow-533061285dd3

Challenges

- Images/drawings/symbols
- Non-text Input
- Color

- Non Capital Letters?
- “However, it still poses difficult challenges for machines to recognize handwritten characters, because of the huge variation in individual writing styles.”

Reference

https://azure.microsoft.com/en-us/resources/cloud-computing-dictionary/what-is-computer-vision#optical-character-recognition

https://aws.amazon.com/what-is/ocr/#:~:text=Optical%20Character%20Recognition%20(OCR)%20is,scan%20as%20an%20image%20file

https://www.ibm.com/topics/computer-vision

End