

# computer vision made easy with kuzco

Frank Hull



# Topics to Cover

- a quick overview of typical computer vision framework
- what is prompt-driven computer vision?
- structured outputs for analysis?
- dive into kuzco as an analysis toolkit
- what's new & coming soon to kuzco



# Typical DL Computer Vision Framework

- Training Computer Vision with TensorFlow
  - steep learning curve
- Training Computer Vision with keras or torch
  - less steep learning curve
- **Pre-Trained Computer Vision Models**
  - least steep learning curve



# Issues Faced by Analysts for Training

- Steep Learning Curve
- Lack of Tensor Knowledge
- Limited Images For Training
- Bad/Limited Labels For Training
- Image Processing For Training & Inference



# Issues Faced by Analysts for Pre-Trained

- Way Easier, You Skipped Training! But...
- Pre-defined Use Case
- Image Processing For Inference



# LLMs are Pre-Trained Models

- LLMs are trained on various data types
- PDFs, Blog Posts, Books, etc. & **Images**
- LLMs are:
  - Pre-Trained Models (DL, Neural Nets)
  - Process Images as Tokens
  - ‘Large’ amount of text & image context for dynamic labelling
- kuzco uses LLMs as pre-trained models



# Prompt-Driven Computer Vision?

- Instead of Training a Neural Net on Images & Labels
- Prompt Engineer an LLM to be a Computer Vision Model
- Role: *You are a specialized AI model functioning as a high-precision Computer Vision engine.*



# Prompt-Driven Computer Vision with kuzco

- kuzco is a collection of system and image prompts for:
  - classification
  - object detection
  - image sentiment
  - alt-text creation
  - text extraction



# Structured Outputs with kuzco

- instead of streams of text, organized as json or tables
- skip dealing with unstructured text and outputs
- kuzco returns all outputs as `data.frames`!



# kuzco as a Computer Vision Toolkit

- we've gone over how kuzco contains a boilerplate of:
  - system + image prompts,
  - structured outputs
- it also contains I/O helpers and a shiny application

```
1 # install the kuzco with pak or devtools in one line:  
2 # via github or r-universe  
3 devtools::install_github("frankiethull/kuzco")
```



# kuzco Input Example

```
1 # load the kuzco library
2 library(kuzco)
3
4
5 my_image = system.file("img/test_img.jpg", package = "kuzco")
6
7 view_image(my_image)
```



# kuzco classification

## Example

- classification

```
1 output = llm_image_classification(  
2     image = my_image,                      # image from the last slide  
3     provider = "ollama",                   # a provider supported by ellmer  
4     llm_model = "qwen2.5vl",               # any model supported by provider  
5     backend = "ellmer"                     # either ellmer or ollamar  
6 )  
7  
8 view_llm_results(output)
```

### LLM Computer Vision



Context	LLM Response
Image Classification	portrait of a dog
Primary Object	puppy
Secondary Object	



# LLM Computer Vision



Context	LLM Response
Image Description	The portrait shows a close-up of a black and white puppy with large brown eyes, looking directly ahead.
Image Colors	[#242424, #8B0000, #F5AF45, #ffffff, #A52A2A]
Image Proba Names	['dog', 'face', 'eyes', 'puppy', 'nose']
Image Proba Values	[0.5, 0.2, 0.15, 0.1, 0.05]



# kuzco Alt-Text Example

- alt-text

```
1 output = llm_image_alt_text(  
2     image = my_image,                      # image  
3     provider = "ollama",                   # a provider supported by ellmer  
4     llm_model = "qwen2.5vl",               # any model supported by provider  
5     backend = "ellmer"                     # either ellmer or ollamar  
6 )  
7  
8 view_llm_results(output)
```

## LLM Computer Vision



Context	LLM Response
Text	A black and white puppy with spots, looking directly at the camera.



# kuzco Recognition Example

- image recognition

```
1 output = llm_image_recognition(  
2     image = my_image,           # image  
3     recognize_object = "nose",   # an object to look for within the image  
4     provider = "ollama",        # a provider supported by ellmer  
5     llm_model = "qwen2.5vl",    # any model supported by provider  
6     backend = "ellmer"          # either ellmer or ollamar  
7 )  
8  
9 view_llm_results(output)
```

## LLM Computer Vision



Context	LLM Response
Object Recognized	TRUE
Object Count	1
Object Description	The image shows a close-up of a dog's face, with a nose that is prominent and black.



# kuzco Shiny App

1 kuzco\_app()

computer vision with LLMs

Select Function

llm\_image\_extract\_text

Backend

ellmer

Model Name

qwen2.5vl

Upload Image

Browse... text\_im

Upload complete

Run LLM

Image Preview

Results

LLM Computer Vision 🦸

Context	LLM Response
Text	Picture of of Odin as a a puppy circa Q4 2019
Confidence Score	0.95



# kuzco Workflows

- modular and useful with other tidy analysis frameworks
- recommend cloud providers for larger tasks
- follow ellmer instructs for API keys

```
1 vision_workflow ← \(img){  
2  
3 # for mistral:  
4 # Sys.setenv(MISTRAL_API_KEY = "the_api_key_via_the_provider")  
5 pixtral_classifier ← kuzco::llm_image_classification(  
6   image = img,  
7   llm_model = "pixtral-12b",  
8   provider = "mistral",  
9   backend = "ellmer"  
10 )
```



```
11  
12  pixtral_sentiment ← kuzco::llm_image_sentiment(  
13      image = img,  
14      llm_model = "pixtral-12b",  
15      provider = "mistral"  
16      backend = "ellmer"  
17  )
```



# New to kuzco

- developer tools
  - kuzco has added a few prompt features over the summer including:  
`llm_image_custom`

```
1 custom_prompt = llm_image_custom(  
2     llm_model = "qwen2.5vl",  
3     image = my_image,  
4     system_prompt = "you are a dog breed expert, you know all about dogs.  
5                     tell me the primary breed, secondary breed, and a brief  
6     image_prompt = "tell me what kind of dog is in the image?",  
7     example_df = data.frame(  
8         dog_breed_primary = "hound",  
9         dog_breed_secondary = "corgi",  
10        dog_breed_information = "information about the primary and se  
11    )  
12 )
```



13

14 view\_llm\_results(custom\_prompt)

## LLM Computer Vision



Context	LLM Response
Dog Breed Primary	Australian Cattle Dog
Dog Breed Secondary	Dalmatian
Dog Breed Information	The primary breed is the Australian Cattle Dog, known for its distinctive liver and white coat and confident, alert demeanor. The secondary breed is the Dalmatian, which is easily recognizable by its black spots over a white coat and a friendly, energetic character.



# Coming Soon

- additional developer tools
  - roadmap includes system and image prompt editing tools
- CRAN release



# Thank You for Attending!

- any questions?

