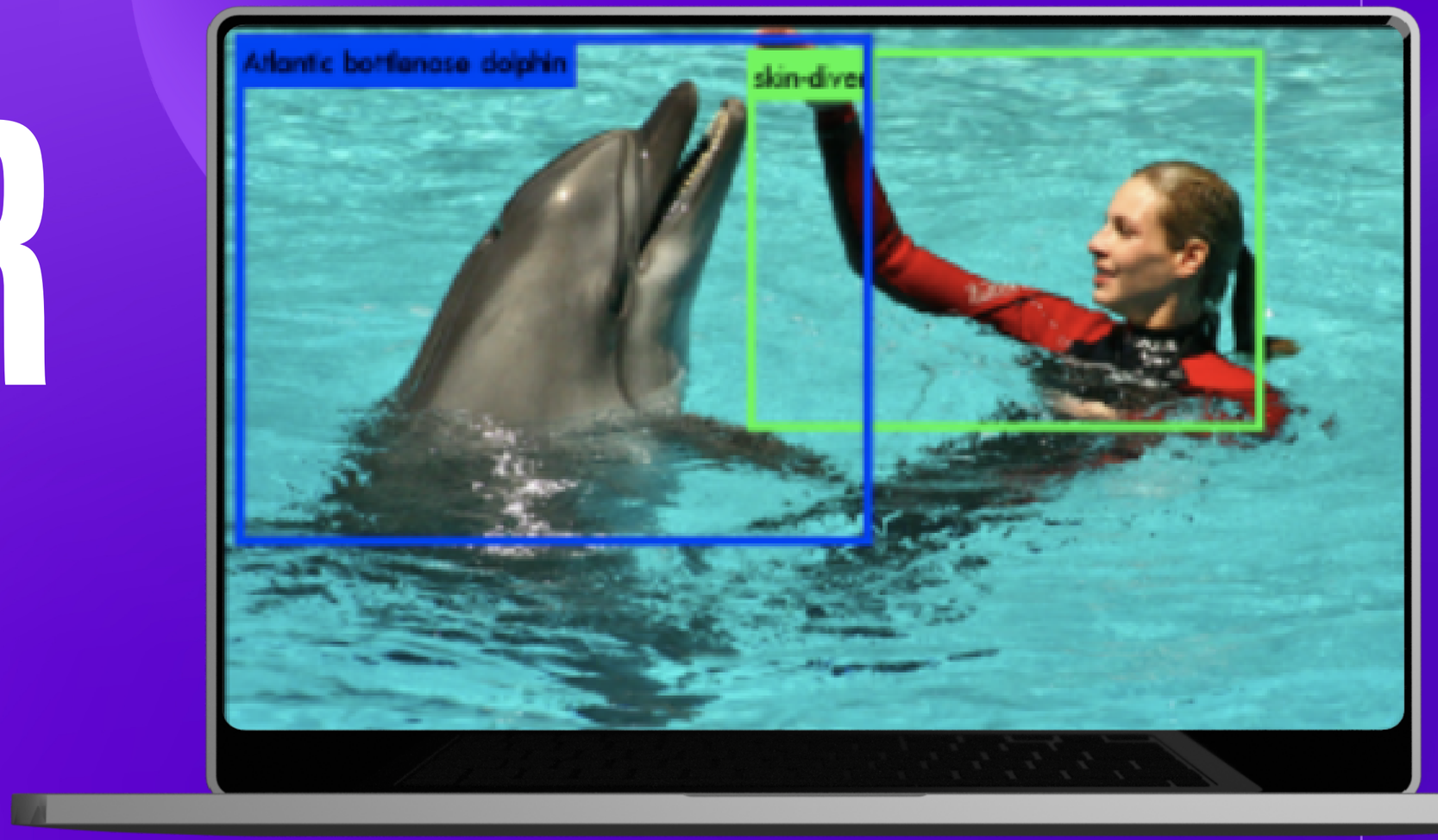
YOLO FAMILY

YOLOR

VISUAL LEARNERS



INTRODUCTION

- We have seen previously that YOLO stands for You Only Look Once, but YoloR is slightly different.
- YOLOR stands for “You Only Learn One Representation”, not to be confused with YOLO versions 1 through 4. It is a state-of-the-art machine learning algorithm for object detection, different from YOLOv1-YOLOv5 due to the difference in authorship, architecture, and model structure.
- At first this does not make much sense, but after reading their paper or this article, you’ll see that the title wraps up what its about, quite nicely!

CREATORS

- YOLOR authors include Chien-Yao Wang, I-Hau, The, and Hong-Yuan Mark Liao.
- However, the previous versions of YOLO implementations have been created by Joseph Redmon and Ali Farhadi (YOLOv1 to YOLOv3), and Aleksey Bochkovskiy (YOLOv4).
- The authors of YOLOR represent the Taiwanese Institute of Information Science, Academia Sinica, and Elan Microelectronics Corporation of Taiwan.

HOW YOLOR WORKS

Humans are able to learn and understand the physical world based on vision or hearing (explicit knowledge) – but also based on past experience (implicit knowledge).

Therefore, humans are able to effectively process entirely new data by making use of abundant experience from prior learning that is gained through normal learning and stored in the brain.

HOW YOLOR WORKS

Based on this idea, the YOLOR research paper describes an approach to combine explicit knowledge, defined as learning based on given data and input, with implicit knowledge learned subconsciously.

Therefore, the concept of YOLOR is based on encoding implicit and explicit knowledge together, similar to how mammalian brains process implicit and explicit knowledge in conjunction with each other.

The unified network proposed in YOLOR generates a unified representation to serve a variety of tasks all at once.

EXPLICIT KNOWLEDGE

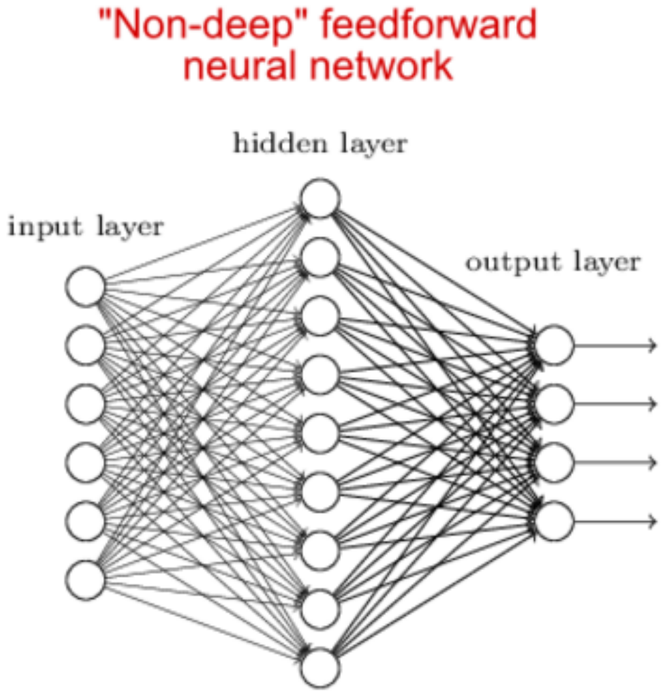
- Explicit knowledge is known as normal learning, or things that you learn consciously. If I walk up to you and say that this is a dog, you will be like aha, that is dog.

- Explicit knowledge is given to neural networks by providing clear metadata or image databases that are either thoroughly annotated or well organized.

- Explicit knowledge can be thought of like flashcards for the machine learning model, with clear definitions and pictures/inputs corresponding to those images.

EXPLICIT KNOWLEDGE

- After the model goes through the deck of flashcards, it is now well-versed in classifying images with their respective definitions, or “classes".
- Explicit knowledge for YOLOR is obtained from the shallow layers of the neural networks.
- This knowledge directly corresponds to observations that are supposed to be made.



EXPLICIT DEEP LEARNING

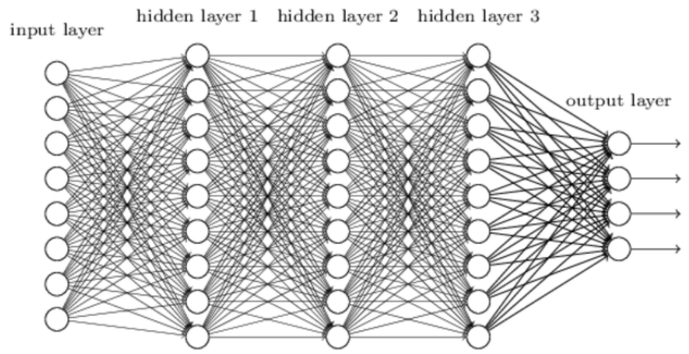
- Explicit Deep Learning was just briefly touched on in the paper but they mentioned that this can be achieved using Detection Transform (DETr), Non-Local Networks and Kernel Selection.

- Remember earlier on we mentioned that Explicit Knowledge is based on observation, well the authors essentially used Scaled YOLOv4 CSP for their explicit model.

IMPLICIT KNOWLEDGE

- Implicit knowledge on the other hand refers to the knowledge learnt subconsciously, sort of like riding a bike or learning how to walk. Its derived from experience.
- Implicit knowledge can be defined as the detailed feature extraction from given data whereas explicit is rough feature extraction.
- For neural networks, implicit knowledge is obtained by features in the deep layers. The knowledge that does not correspond to observations is known as implicit knowledge as well.

Deep neural network



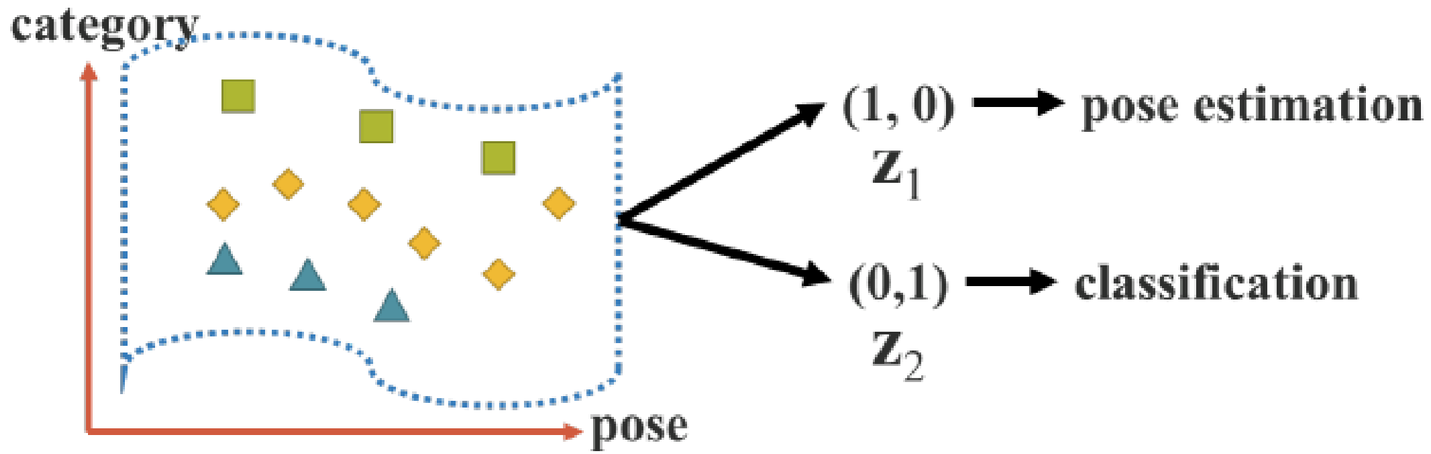
IMPLICIT DEEP LEARNING

There’s a couple of ways in which we can implement Implicit Knowledge. These include:

- Manifold Space Reduction,
- Kernel Alignment, and
- More Functions.

MANIFOLD SPACE REDUCTION

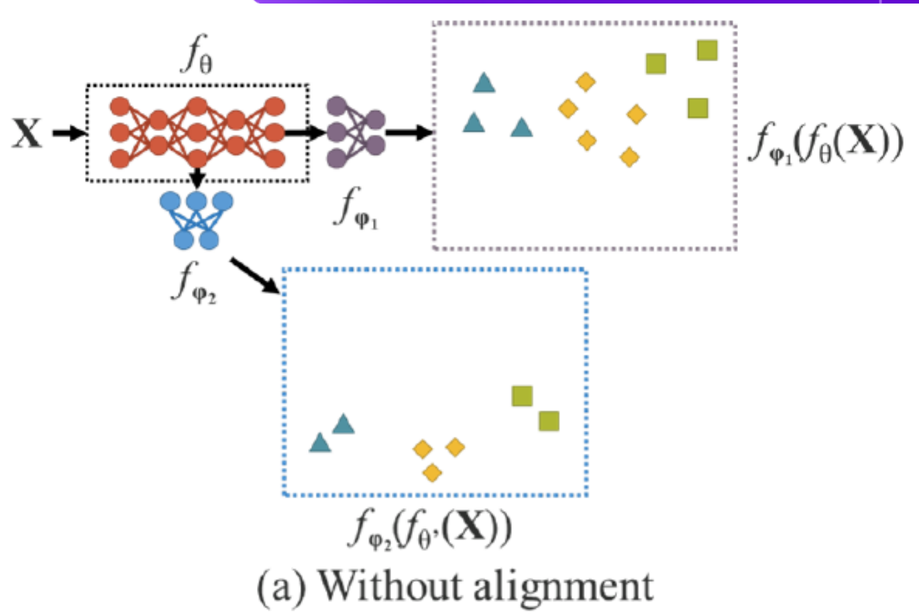
For manifold space reduction, my understanding is that we reduce the dimensions of the manifold space so that we are able to achieve various tasks such as pose estimation and classification, amongst others.



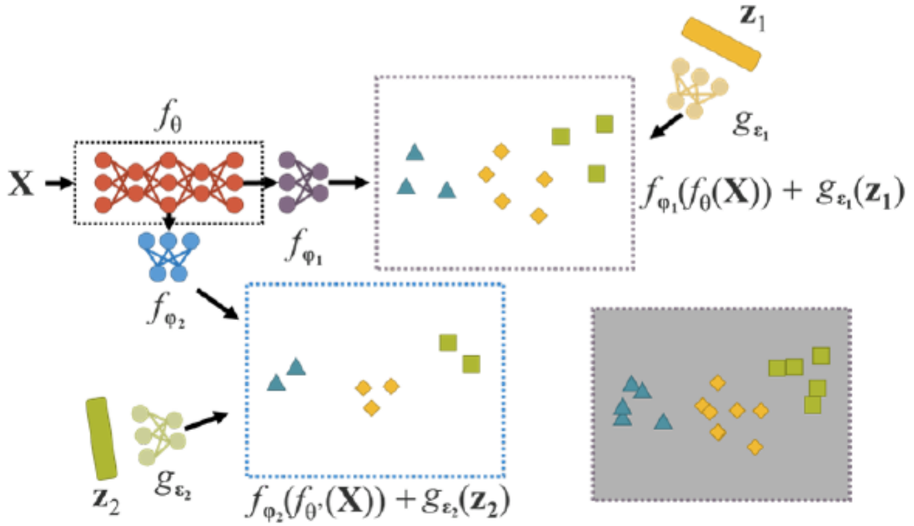
KERNEL SPACE ALIGNMENT

- So they mention that in multi-task and multi-head neural networks, kernel space misalignment is a frequent problem.

- In order to deal with this problem, they perform both addition and multiplication of the output feature and implicit representations, so that the Kernel can be translated, rotated and scaled in such a way as to align each output kernel of a network.



(a) Without alignment

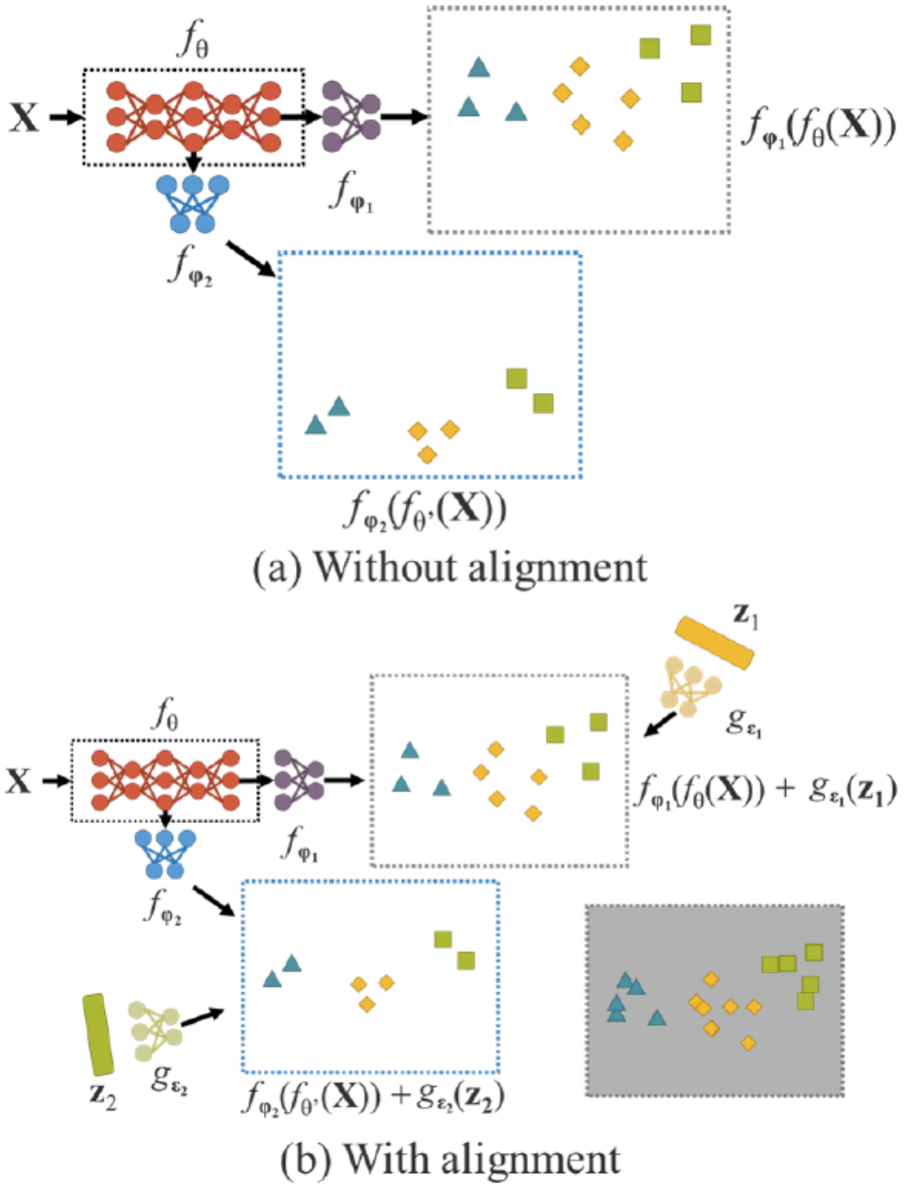


(b) With alignment

KERNEL SPACE ALIGNMENT

- What they mean is that this is essentially important for aligning features of large and small objects in Feature Pyramid Networks (FPN).

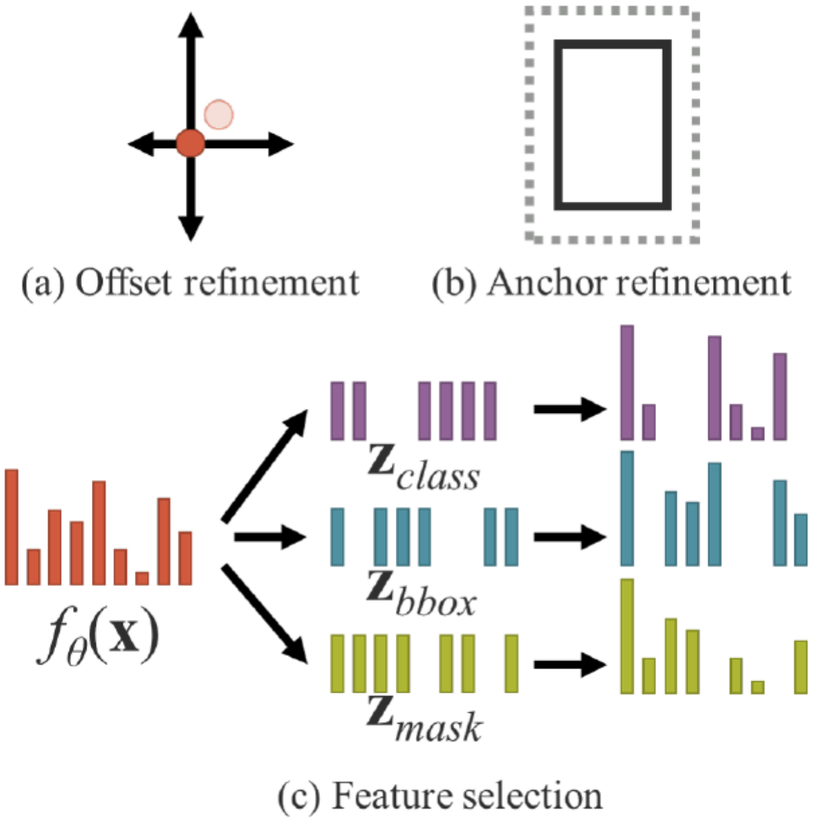
- You can imagine the amount of visual differences or disparity there are in pixels between objects that are far, to those that are close to your camera.



MORE FUNCTIONS

- In addition to the aforementioned techniques, they propose that you can also apply some operations for offset and anchor refinement as well as feature selection.

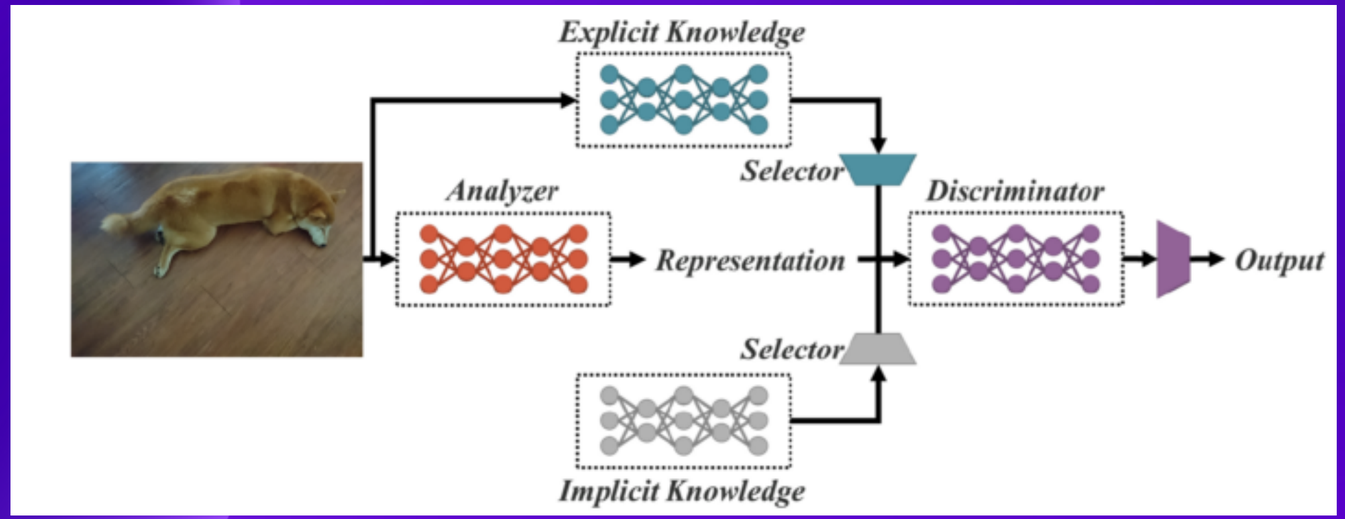
- Meaning that you can apply different operations to perform different tasks like either getting the class of the object, the bounding box or the mask amongst many other potential tasks .



THE UNIFIED MODEL

To recap, features obtained from shallow layers are known as **“explicit knowledge,”** while features obtained from deep layers are called **implicit knowledge**.

A unified network is a network that is a combination of both Implicit and Explicit knowledge.



THE UNIFIED MODEL

The authors combined both explicit and implicit knowledge in a unified network that can accomplish various tasks.

Early on in the paper they do not mention what these tasks are, but if you browse through the experimentations section, they mention that they plan to do the following tasks like Object Detection, Instance Segmentation, Panoptic Segmentation, Keypoint detection, image captioning and more...

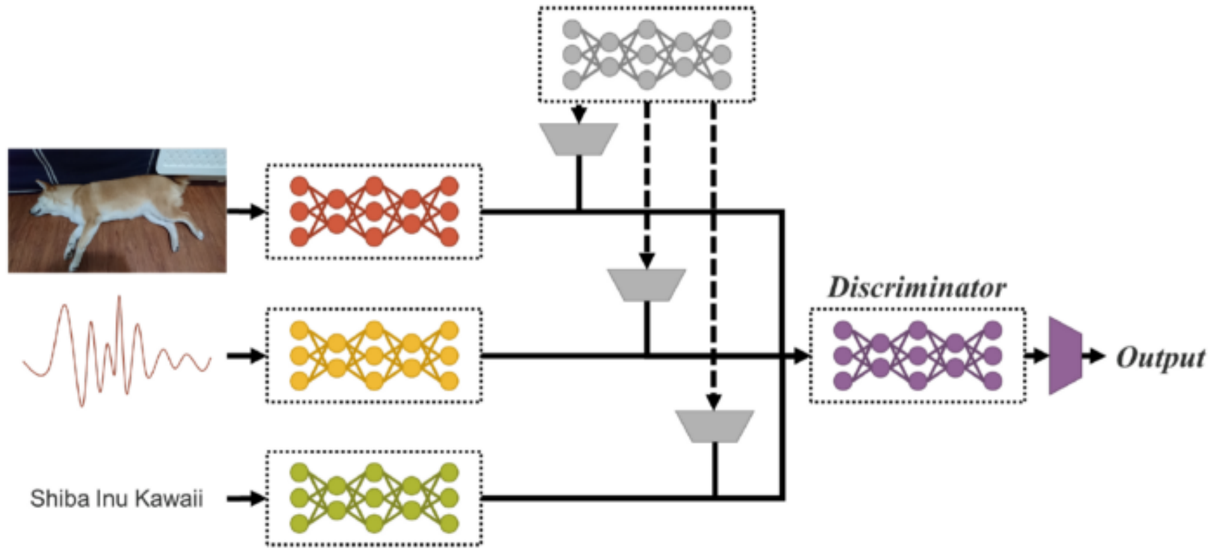


Figure 12: Multimodal unified netwrok.

INTEGRATION OF IMPLICIT AND EXPLICIT LEARNING

This leads us up to applying Implicit knowledge into the Unified Network. If we had to model a Conventional Network it would look something like this:

$$y = f_\theta(\mathbf{x}) + \epsilon$$

where x is the observation, for example, you see a dog.

Theta is the set of parameters of a neural network,

f_**θ** represents the operation of the neural network, and

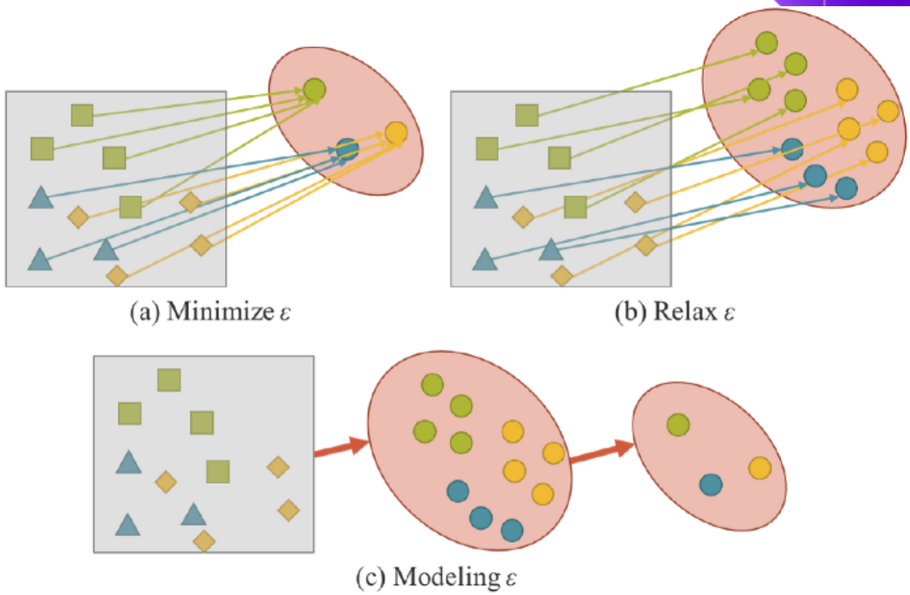
ϵ is the error term.

CONVENTIONAL NETWORK PROBLEM

- The authors state that when one trains a conventional neural network, the goal is to minimize the error and to make f(x) with respect to **θ** as close to the target y as possible.

- But now if we want to perform various other tasks like **object segmentation** and **pose estimation** for example, we would have to relax the error term to find the solution space for each task.

- This currently is quite challenging to do, but a solution that the authors proposed is to rather model the error term **ϵ** to find solutions for various tasks as you can see here.



UNIFIED NETWORKS

- This is where unified networks come into the picture, we can now expand our equation to incorporate the implicit model with g theta and the explicit error from observation x together with the implicit error from z which they term the latent code.

$$y = f_\theta(\mathbf{x}) + \epsilon + g_\phi(\epsilon_{ex}(\mathbf{x}), \epsilon_{im}(\mathbf{z}))$$

- In layman's terms it just refers to the representation of compressed data that makes up the implicit knowledge.

UNIFIED NETWORKS

- We can further simplify the equation to this.

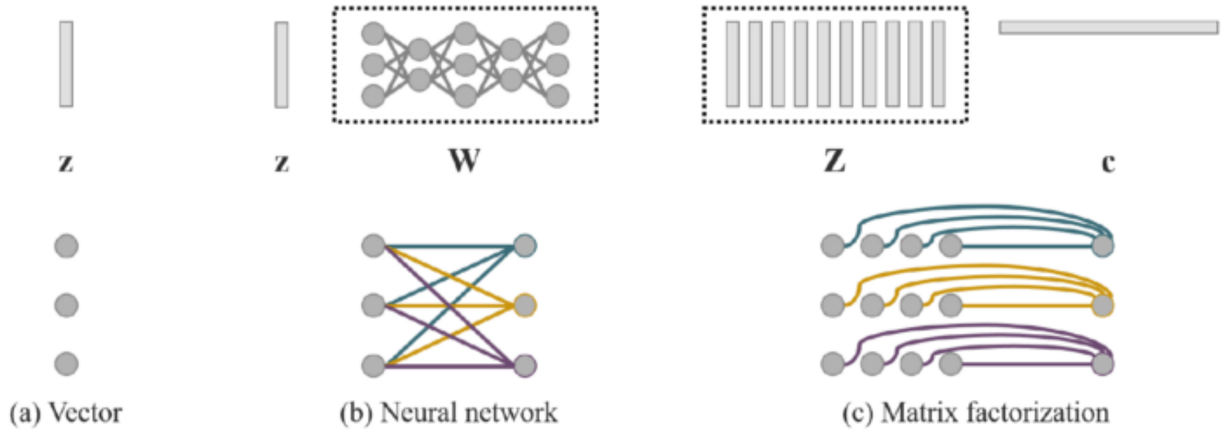
$$y = f_\theta(\mathbf{x}) \star g_\phi(\mathbf{z})$$

- Where the $\star$ represent some possible operators like addition or concatenation that combines f and g or, rather the explicit with the implicit models.

MODELLING IMPLICIT KNOWLEDGE

There’s a lot more more mathematics involved with the unified network architecture but lets now move on to how Implicit Knowledge can actually be modelled, which is in three ways:

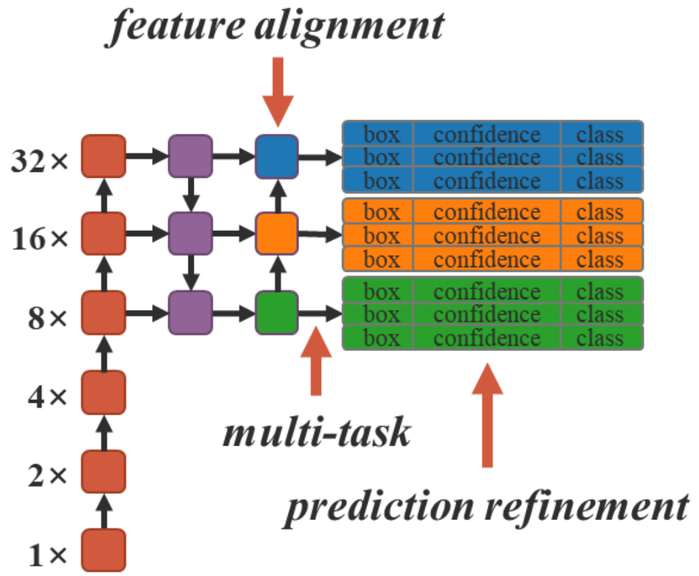
- As a Vector,
- A Neural Network, or as a
- A Matrix.



YOLOR ARCHITECTURE

The architecture of this model YOLOR proposes a single neural network to apply implicit knowledge to three aspects including

- Feature alignments on Feature Pyramid Networks (FPN)
- Prediction Refinement, and
- Multi-task learning in a single model.

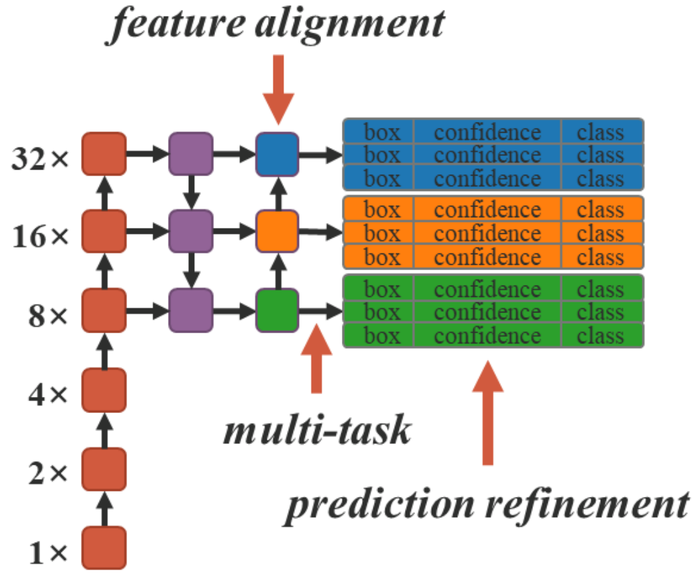


YOLOR ARCHITECTURE

- Multi task learning includes performing tasks like detecting objects, and multi level image classification and Feature Embedding.

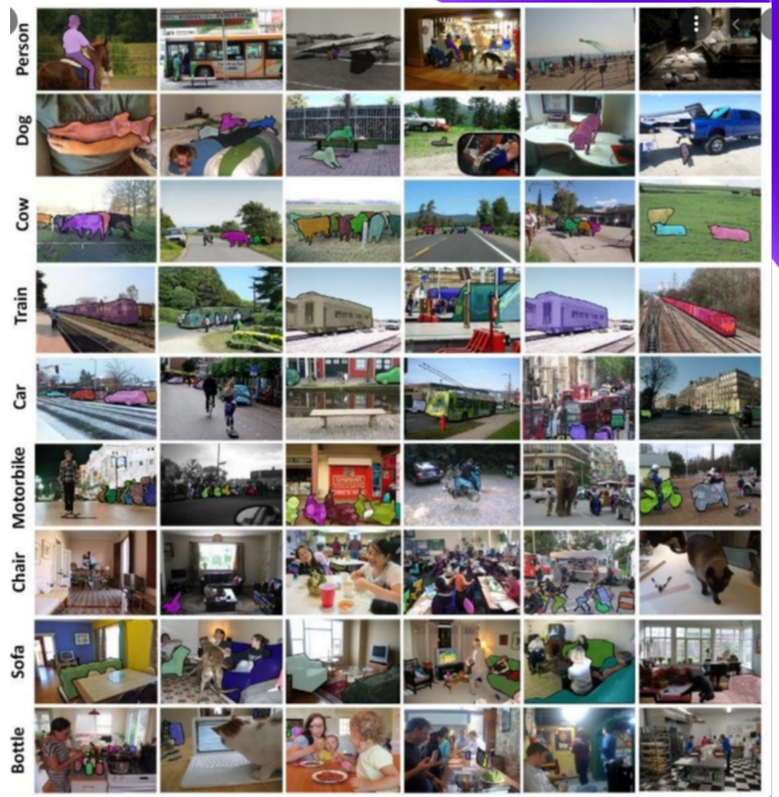
- Feature embedding refers to the process of extracting features and classifying them based on the attributes of these features.

- Prediction refinement on the other hand enhances the model by performing correction using loss function.



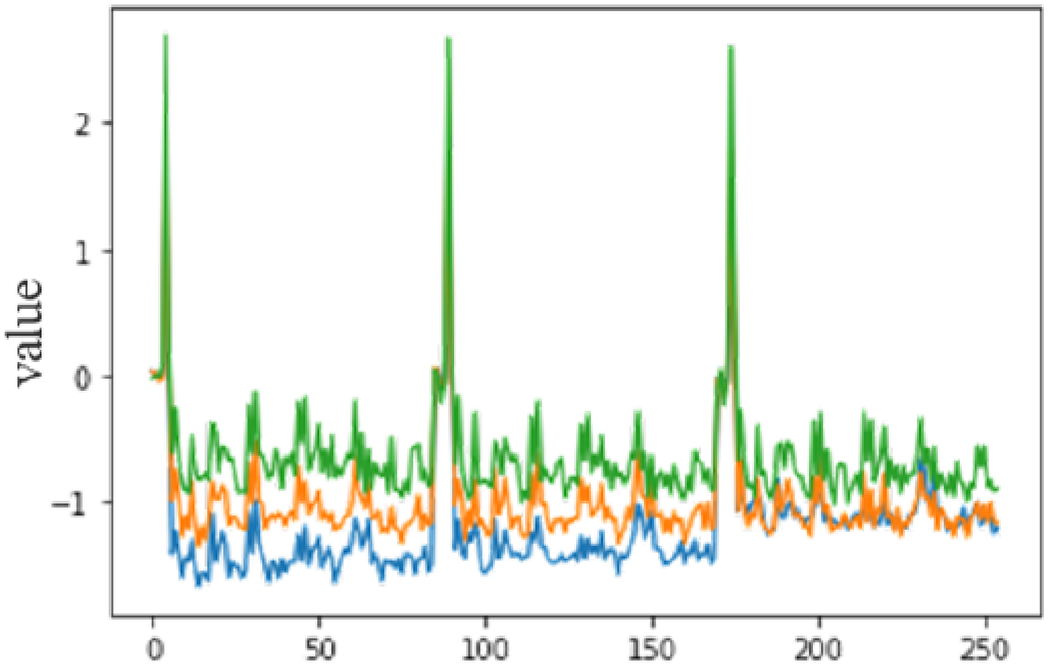
TRAINING

- They use YOLOV4-CSP as their base line model and introduced implicit knowledge into the model and trained it on the MS-COCO Dataset.
- The way they tested the unified model was through Ablation Study, which is just a fancy-ass way to say that they are testing one method at a time to understand the contribution of the component to the overall system.



TRAINING

They followed the same training process as scaled-yolov4 in which they trained, from scratch, 300 epochs first and then fine tuned 150 epochs thereafter.



dim=255: $[(x, y), (w, h), obj, 80\ classes] * 3$

RESULTS

- If you look at the baseline, you can see the effect of the implicit model compared to the baseline.
- You can see that in terms of accuracy YoloR is comparable but where it shines, is in its frame rate. Its almost double the frame rate of Scaled YOLOv4

Table 9: Comparion of state-of-the-art.

Method	pre.	seg.	add.	AP^{test}	AP^{test}_{50}	AP^{test}_{75}	FPS^{V100}
YOLOR (ours)				55.4%	73.3%	60.6%	30
ScaledYOLOv4 [15]				55.5%	73.4%	60.8%	16
EfficientDet [13]	✓			55.1%	74.3%	59.9%	6.5
SwinTransformer [10]	✓	✓		57.7%	–	–	–
CenterNet2 [26]	✓		✓	56.4%	74.0%	61.6%	–
CopyPaste [6]	✓	✓	✓	57.3%	–	–	–

RESULTS

If you look at the test run by the legend Alexey Bochkovskiy, they show an 88% in improvement in speed when compared to Scaled YOLOv4 and a 3.8% improvement in Average Precision compared to PP-Yolov2.