Could you explain YOLO working on an example like we an image containing dog and cat , we have divided the image into 3*3 grids cells , now what will each cell does for each small contribution by containing small part of dog or cat and suppose some grid cells have not dog nor cat , i want to know how each cell will behave corresponding to what it containing and also explain how IOU and NMS will work here.

Perfect question Salman — this is *exactly* how you should think to understand YOLO deeply. Let's go step by step through the **entire process** using your example — an image containing a **dog and a cat**, divided into a **3×3 grid**.

□ Step 1: Divide the image into grid cells

We divide the image into **3×3 = 9 grid cells** (each cell is responsible for detecting objects whose **center** lies inside it).

Each grid cell will output:

- Bounding box coordinates: `(x, y, w, h)`
- Confidence score: how confident the model is that an object exists in that box.
- Class probabilities: which class it might be (dog, cat, etc.)

So, each cell predicts something like this:

```
[objectness score, x, y, w, h, P(dog), P(cat)]
```

□ Step 2: What each grid cell "sees"

Now let's imagine this scene:

Cell	Contains part of	Center of object inside?	Action
(1,1)	background (sky)	□	predicts "no object"
(1,2)	ear of dog	□	"no object" (just part, not center)
(2,2)	**dog's face center** □	□	predicts "object exists (dog)"
(2,3)	cat tail	□	no object
(3,1)	ground	□	no object
(3,2)	**cat's body center** □	□	predicts "object exists (cat)"
others	background	□	no object

□ **Key idea**:
YOLO assigns **only one object per grid cell** — specifically, the object whose **center** lies inside it.

So:

- Cell (2,2) is responsible for predicting the **dog's bounding box**.
- Cell (3,2) is responsible for predicting the **cat's bounding box**.

□ Step 3: What each cell predicts numerically

For example,
Cell (2,2) might predict:

```
objectness = 0.9
x = 0.4, y = 0.5, w = 0.6, h = 0.7
P(dog)=0.95, P(cat)=0.05
```

Cell (3,2) might predict:

```
objectness = 0.88
x = 0.5, y = 0.6, w = 0.4, h = 0.5
P(dog)=0.1, P(cat)=0.9
```

Other cells might predict very low objectness (e.g. 0.05), meaning "no object".

□ Step 4: Combine scores

For each box:

```
Final score = objectness × class probability
```

So:

- Dog box score = 0.9 × 0.95 = **0.855**
- Cat box score = 0.88 × 0.9 = **0.792**

These are the **confidence scores** used later in filtering.

□ Step 5: Multiple overlapping predictions (repetition)

Sometimes, **neighboring cells** may predict **slightly overlapping boxes** for the same object (because part of the dog appears in multiple cells).
Example:

Cell	Predicted IOU with GT (dog)	Confidence
(2,2)	0.9	0.85
(2,3)	0.6	0.7
(1,2)	0.4	0.3

Now we have **repetitions** → multiple boxes for the same dog.

□ Step 6: IOU (Intersection over Union)

IOU = area of overlap / area of union

It measures how much two boxes overlap.

- IOU = 1 → perfect overlap
- IOU = 0 → no overlap

We use IOU to compare:

1. Predicted boxes vs Ground Truth (during training)
2. Predicted boxes vs other predicted boxes (during NMS filtering)

In our case:

- (2,2) box has high IOU with the ground truth dog box → it's a good prediction.
- (2,3) box has medium IOU → less accurate.
- (1,2) box has low IOU → poor prediction.

□ Step 7: Non-Max Suppression (NMS)

NMS removes **duplicate bounding boxes** for the same object.

Here's how it works for dog boxes:

1. Sort all dog predictions by confidence:

```
2. (2,2): 0.85
3. (2,3): 0.7
4. (1,2): 0.3
```

5. Pick the **highest confidence** box → (2,2).
6. Compare this box with the others:
    - If IOU > threshold (say 0.5), remove those boxes (they're duplicates).

So:

- (2,3) removed (IOU=0.6 > 0.5)
- (1,2) kept (IOU=0.3 < 0.5)

Result: only (2,2) box remains for the dog □

Same thing happens for the **cat boxes**.

□ Step 8: Final Output

After NMS, YOLO keeps:

```
1. Dog → box from cell (2,2)
2. Cat → box from cell (3,2)
```

Bounding boxes drawn:

- Tight around the dog and cat.
- Clean, no overlaps or duplicates.

□ Summary of each concept

Concept	Meaning	Role
Grid Cell	Splits image into regions	Each cell predicts for object whose center is inside
Bounding Box (x, y, w, h)	Coordinates of detected object	Localizes object
Objectness Score	Probability that an object exists	Filters empty cells
Class Probabilities	Prob. of each class (dog/cat)	Helps classify
IOU	Overlap measure	Used to evaluate and filter
NMS	Removes duplicate boxes	Keeps best prediction per object

□ Visualization of Behavior

Imagine this (simplified):

```
+-----+-----+-----+
|Sky  | Dog | Dog |
+-----+-----+-----+
|Grass|DogC.| Cat |
+-----+-----+-----+
|Ground|CatC.|Cat |
+-----+-----+-----+
```

- "DogC." = center of dog → cell predicts the dog
- "CatC." = center of cat → cell predicts the cat
- All others predict background

After IOU + NMS → only two bounding boxes remain:

- Dog box
- Cat box