Enhanced Bot Game Using Object Detection

Present by

6220421001 Chayakan Chanchai
6220421007 Manadda Jaruschaimongkol
6310421012 Sarawut Kodcha

Because...

Some players **don't have much time** to play games

Some players decide to **spend the money** to hire other players to play

Some players feel **bored** and stop playing the game

some players **use macro programs** to control and facilitate the game but it is not so good bot

- Detected as a bot by a game system and banned
- Cheated
- The number of players has decreased
- Revenue has decreased
- Game closed down

To attract new players, retain existing players, and make better experiences

an enhanced bot game will help them customize their playing styles

Scope of Work

- Enhanced a bot using object detection instead of a macro program
- Detecting only objects that players can interact with
- Classifying objects to predict an action

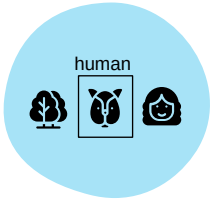
Limitations

- Focusing on objects in one area only
- It's just a model that will help develop bots to work better

Work Process

Data Preparation

- Screen Captur
- Annotation
- Augmentation



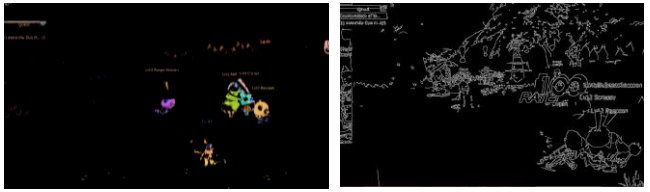
Object Detection

- Match Template
- HSV Thresholding
- Canny Edge
- Cascade Classifier
- YOLOv5

Model Evaluation

- Performance
- mAP
- Precision
- Recall

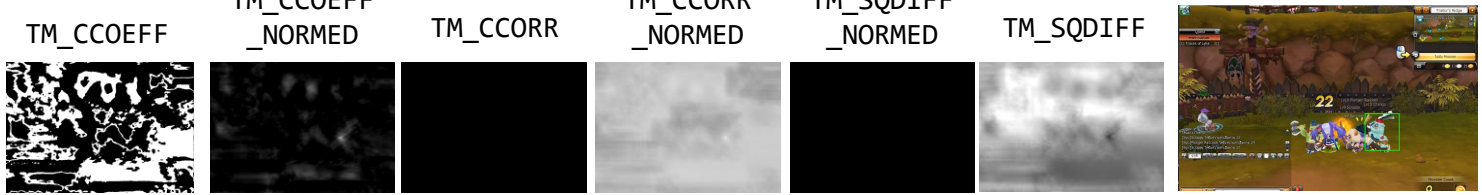
Finding the training images and preprocessing the data

	Input data	Image Annotation and Augmentation	Final Input Data
Match Template	Template + Detected Object	-	
HSV Thresholding, Canny Edge	Detect by creating Tracker	-	
Cascade Classifier	Positive x 600 img Negative x 400 img	1 class OpenCV - Positive x 600 img OpenCV - augmentation x 1,400 img	 + Text file specifying no. object and location of annotated object
YOLOv5	Training x 606 img	6 classes LabelImg - annotation x 606 img Roboflow - augmentation x 1,212 img	 + Text file specifying object class and location of each annotated object

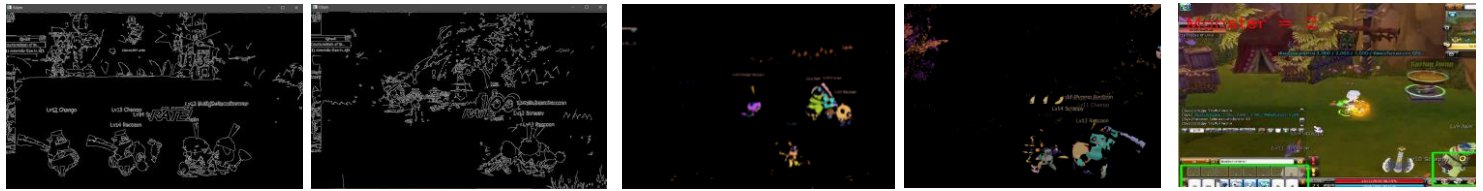
Then perform object detection

Match Template

TM_CCOEFF | TM_CCOEFF_NORMED | TM_CCORR | TM_CCORR_NORMED | TM_SQDIFF_NORMED | TM_SQDIFF



HSV Thresholding, Canny Edge



Cascade Classifier

20 stage;pos 2.6k img; neg 1.2k img; time 4hrs | 10 stage; pos 2.0k img; neg 2.0k img; time 4hrs 30mins

YOLOv5

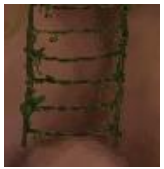
6 classes, 1.8k img, 100 epoch, batch size 16, time 23 mins

In parallel, we train the YOLOv5 model to classify objects based on players' action

We train **6 classes based** on players' action

Character

No action



Stair

Jump

C + ⬆

Monster

Kill

Z

or set skill button

NPC

Get quest

Space Bar

Spring

Jump

C

Box

Jump

C

Custom **Training**

- Create labels

```
0 0.569102 0.802152 0.123396 0.336093
1 0.637710 0.758278 0.057256 0.175497
0 0.793189 0.776490 0.127345 0.301325
```

```
monster
character
npc
box
```

- Model Selection > **YOLOv5s**

we chose the smallest, fastest base model of YOLOv5

Small	Medium	Large	XLarge
YOLOv5s	YOLOv5m	YOLOv5l	YOLOv5x
14 MB$_{FP16}$	41 MB$_{FP16}$	90 MB$_{FP16}$	168 MB$_{FP16}$
2.0 ms$_{V100}$	2.7 ms$_{V100}$	3.8 ms$_{V100}$	6.1 ms$_{V100}$
37.2 mAP$_{COCO}$	44.5 mAP$_{COCO}$	48.2 mAP$_{COCO}$	50.4 mAP$_{COCO}$

- Model Training > **batch = 16; epochs = 100**

Here are our results, the best technique is YOLOv5

Match Template

Accurate but not dynamic

Cascade Classifier

Poor Detection, Worst Performance

VDO Link: https://mega.nz/file/2hdWWRRZ#grHh7P9044QrjzmFlnTdpiKa8me55AYoxBmhb5c8ie4

HSV Thresholding, Canny Edge

Poor Detection, Poor Performance

VDO Link: https://mega.nz/file/3w1VSS6R#51oKMwR7KHF3sVcUDbn0ag3F9QUQ3tLg2JmiMQu1yII

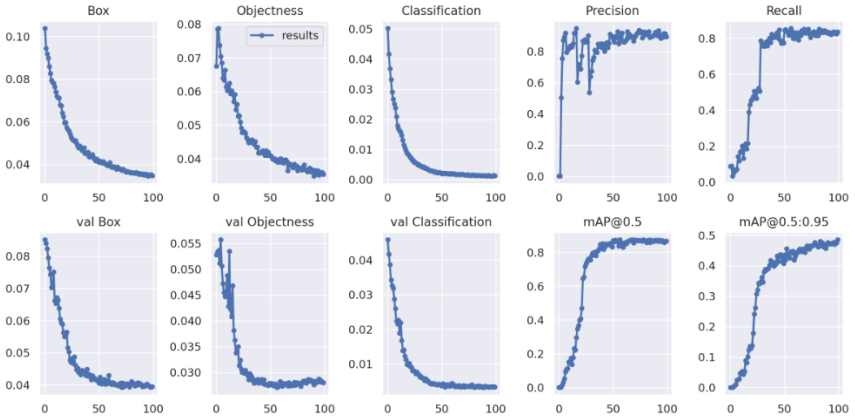
YOLOv5

Best Detection, Good Performance

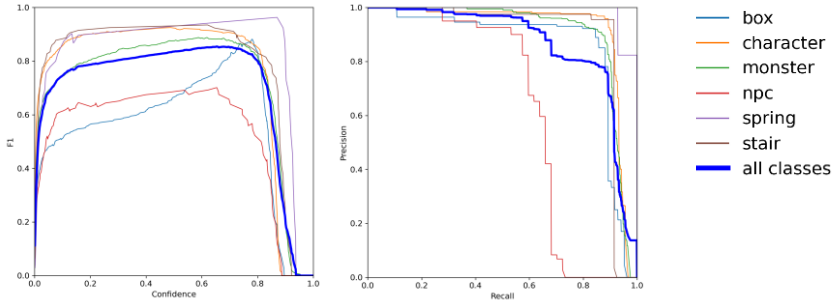
VDO Link: https://mega.nz/file/24NgiAZL#3M9_B7XN5TpjUe9_peDzL2Bo37Y-sPfeZ-_Pj7ha2Sk

YOLOv5 Model Evaluation

Training result



Training result each class



Result from last weight on Traning Set

Class	Images	Targets	P	R	mAP@.5	mAP@.5:.95:
all	1.04e+03	7.13e+03	0.969	0.962	0.981	0.663
box	1.04e+03	810	0.931	0.899	0.961	0.558
character	1.04e+03	1.07e+03	1	0.999	0.996	0.69
monster	1.04e+03	4.64e+03	0.979	0.963	0.99	0.696
npc	1.04e+03	150	0.934	0.947	0.963	0.572
spring	1.04e+03	93	0.991	0.968	0.98	0.722
stair	1.04e+03	366	0.978	0.994	0.994	0.741

Result from best weight on Traning Set

Class	Images	Targets	P	R	mAP@.5	mAP@.5:.95:
all	1.04e+03	7.13e+03	0.969	0.962	0.981	0.663
box	1.04e+03	810	0.931	0.899	0.961	0.558
character	1.04e+03	1.07e+03	1	0.999	0.996	0.69
monster	1.04e+03	4.64e+03	0.979	0.963	0.99	0.696
npc	1.04e+03	150	0.934	0.947	0.963	0.572
spring	1.04e+03	93	0.991	0.968	0.98	0.722
stair	1.04e+03	366	0.978	0.994	0.994	0.741

Evaluate on Validation Set using the best weight

Class	Images	Targets	P	R	mAP@.5	mAP@.5:.95:
all	130	801	0.894	0.834	0.866	0.487
box	130	84	0.672	0.893	0.861	0.38
character	130	207	0.967	0.86	0.922	0.508
monster	130	402	0.911	0.856	0.907	0.493
npc	130	47	0.9	0.574	0.626	0.253
spring	130	14	0.96	0.929	0.983	0.626
stair	130	47	0.954	0.891	0.898	0.662

Evaluate on Test Set using the best weight

Class	Images	Targets	P	R	mAP@.5	mAP@.5:.95:
all	130	749	0.895	0.842	0.893	0.458
box	130	39	0.762	0.819	0.853	0.355
character	130	189	0.9	0.856	0.908	0.478
monster	130	413	0.916	0.862	0.922	0.526
npc	130	60	0.842	0.633	0.731	0.343
spring	130	26	0.996	0.885	0.989	0.504
stair	130	22	0.955	1	0.958	0.54

Try with Blind Dataset

VDO Link: https://mega.nz/file/7pcAQT4b#49mWnFL1Ch1wwo4NgxfkWDsKjUHP_oumuPb6PeQ2VeQ

Evaluate on Blind Dataset

Class	Images	Targets
all	100	273
box	100	98
character	100	112
monster	100	63

Class	P	R
all	0.0246	0.148
box	0.047	0.296
character	0.0219	0.0536
monster	0.00487	0.0952

Class	mAP@.5	mAP@.5:.95:
all	0.00909	0.00337
box	0.0205	0.00816
character	0.00472	0.00129
monster	0.00207	0.000666

Cannot perform object detection as well on blind dataset

A bot game using object detection is the first step to make better experiences for players

Help players find an object that they want to interact with

With accurate detection to perform best action

Get what they wants without wasting time

The next step is...

Train more objects in different areas and use instance object detection to improve occlusion detection

Develop more feature e.g. total item gain, blood guage, etc.

Implement model and build a bot