Project feasibility

Market Research

Project: DNSS

Prepared by: JTM ABDULLAH KHALID

Phase 1: Research and Conceptualization - Conduct extensive market research to identify emerging trends and user needs. Define a set of artificial features, such as image recognition, augmented reality, and intelligent scene analysis, that align with consumer demands.

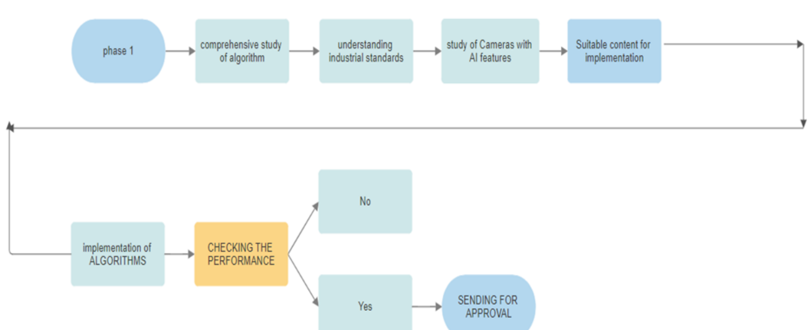
Phase 2: Design and Prototyping - Develop a detailed design incorporating the chosen artificial features. Create a functional prototype to test the integration of these features, ensuring seamless performance and user-friendly interfaces.

Phase 3: Testing and Optimization - Rigorous testing to evaluate the prototype's reliability, accuracy, and overall performance. Collect user feedback to identify areas for improvement, and optimize the camera's algorithms and software for enhanced functionality and responsiveness.

Phase 4: Manufacturing and Assembly - Scale up production based on successful prototype testing. Collaborate with manufacturing partners to ensure mass production meets quality standards while maintaining cost-effectiveness.

PHASE 1	**A.I features on pc**
PHASE 2	**Camera + platform**
PHASE 3	**Gimbal addition**
PHASE 4	**Multicamera addition**

Presented to: Dr. Aamir Irshad



Available resources.

Recommendations for new processors will be issued after comprehensive study.

Features

<table>
<tr><th>Model</th><th>ROCK (PI) 4A</th><th>ROCK (PI) 4B</th><th>ROCK (PI) 4C</th><th>ROCK (PI) 4A Plus</th><th>ROCK (PI) 4B Plus</th><th>ROCK 4 SE</th><th>ROCK (PI) 4C Plus</th></tr>
<tr><td>Processor</td><td colspan="3">64bits hexa core processor
Rockchip RK3399
Dual Cortex-72,
frequency 1.8GHz with qual Cortex-A53, frequency 1.4GHz
Mali T860MP4 gpu, support OpenGL ES 1.1/2.0/3.0/3.1/3.2, Vulkan 1.0, Open CL 1.1 1.2, DX11.</td><td colspan="2">64bits hexa core processor
Rockchip OP1
Dual Cortex-72,
frequency 2.0GHz with qual Cortex-A53, frequency 1.5GHz
Mali T860MP4 gpu, support OpenGL ES 1.1/2.0/3.0/3.1/3.2, Vulkan 1.0, Open CL 1.1 1.2, DX11.</td><td colspan="2">64bits hexa core processor
Rockchip RK3399-T
Dual Cortex-72,
frequency 1.5GHz with qual Cortex-A53, frequency 1.0GHz
Mali T860MP4 gpu, support OpenGL ES 1.1/2.0/3.0/3.1/3.2, Vulkan 1.0, Open CL 1.1 1.2, DX11.</td></tr>
<tr><td>Memory</td><td colspan="7">LPDDR4
64bit dual channel LPDDR4@3200Mb/s, 1GB/2GB/4GB optioal</td></tr>
<tr><td>Storage</td><td colspan="3">eMMC connector
μSD card (μSD slot supports up to 256 GB μSD card)
M.2 SSD (M.2 connector supports up to 2T M2 NVME SSD)</td><td colspan="2">on board eMMC with up to 128GB variant available
μSD card (μSD slot supports up to 256 GB μSD card)
M.2 SSD (M.2 connector supports up to 2T M2 NVME SSD)</td><td>eMMC connector
μSD card (μSD slot supports up to 256 GB μSD card)
M.2 SSD (M.2 connector supports up to 2T M2 NVME SSD)</td><td>eMMC connector
μSD card (μSD slot supports up to 256 GB μSD card)</td></tr>
<tr><td>Display</td><td colspan="2">Standard HDMI 2.0 up to 4k@60
MIPI DSI 2 lanes via FPC connector
HDMI and MIPI DSI can work at the same time, support mirror mode or extend mode.</td><td>Mini DP up to 1440P@60
Micro HDMI 2.0 up to 4k@60
MIPI DSI 2 lanes via FPC connector
HDMI and DP can work at the same time.</td><td colspan="3">Standard HDMI 2.0 up to 4k@60
MIPI DSI 2 lanes via FPC connector
HDMI and MIPI DSI can work at the same time, support mirror mode or extend mode.</td><td>One Micro HDMI 2K up to 1440P@60
One Micro HDMI 4K 2.0 up to 4k@60
MIPI DSI 4 lanes via FPC connector
Only two of HDMI 2k, HDMI 4K and MIPI DSI can work at the same time.</td></tr>
<tr><td>Audio</td><td colspan="6">3.5mm jack with mic
HD codec that supports up to 24-bit/96KHz audio.</td><td>3.5mm jack
HD codec that supports up to 24-bit/96KHz audio.</td></tr>
<tr><td>Camera</td><td colspan="6">MIPI CSI 2 lanes via FPC connector, support up to 800 MP camera(1mm pitch connector).</td><td>MIPI CSI 2 lanes via FPC connector, support up to 800 MP camera(0.3 mm pitch connector).</td></tr>
</table>

Wireless	None	802.11 ac wifi BT 5.0 with on board antenna	None	802.11 ac wifi BT 5.0 with on board antenna	802.11 ac wifi BT 5.0 with external antenna
USB	USB 3.0 OTG X1, hardware switch for host/device switch, upper one USB 3.0 HOST X1, dedicated USB 3.0 channel, lower one USB 2.0 HOST X2				
Ethernet	GbE LAN	GbE LAN with Power over Ethernet (PoE) support additional HAT is required for powering from PoE	GbE LAN	GbE LAN with Power over Ethernet (PoE) support additional HAT is required for powering from PoE	
IO	40-pin expansion header 2 x UART 2 x SPI bus 3 x I2C bus 1 x PCM/I2S 1 x SPDIF 2 x PWM 1 x ADC 6 x GPIO 2 x 5V DC power in 2 x 3.3V DC power in				
Others	RTC RTC battery connector for time backup(optional)				RTC None
Power	USB PD, support USB Type C PD 2.0, 9V/2A, 12V/2A. Qualcomm® Quick ChargeTM: Supports QC 3.0/2.0 adapter, 9V/2A, 12V/1.5A				USB C 5V/3A
Size	85mm x 54mm				

PHASE 1

Comprehensive study of algorithm

link of paper

http://pjreddie.com/yolo/

You Only Look Once: Unified, Real-Time Object Detection

(YOLO V8)

We unify the separate components of object detection into a single neural network. Our network uses features from the entire image to predict each bounding box. It also predicts all bounding boxes across all classes for an im- age simultaneously. This means our network reasons glob- ally about the full image and all the objects in the image. The YOLO design enables end-to-end training and real- time speeds while maintaining high average precision.

Our system divides the input image into an $S \times S$ grid. If the center of an object falls into a grid cell, that grid cell is responsible for detecting that object.

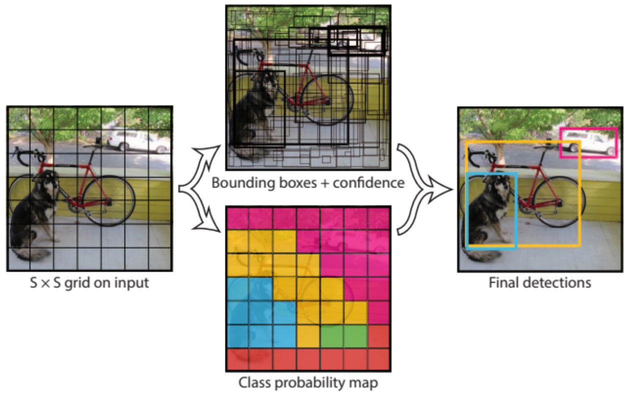
Each grid cell predicts B bounding boxes and confidence scores for those boxes. These confidence scores reflect how confident the model is that the box contains an object and also how accurate it thinks the box is that it predicts. For-escoresshouldb zero. Otherwise we want the confidence score to equal the intersection over union (IOU) between the predicted box and the ground truth.

Each bounding box consists of 5 predictions: x, y, w, h, and confidence. The (x, y) coordinates represent the center of the box relative to the bounds of the grid cell. The width and height are predicted relative to the whole image. Finally the confidence prediction represents the IOU between the predicted box and any ground truth box.

Each grid cell also predicts C conditional class proba- bilities, $\Pr(\text{Class}_i|\text{Object})$. These probabilities are condi- tioned on the grid cell containing an object.

$$\Pr(\text{Class}_i|\text{Object}) \ast \Pr(\text{Object}) \ast \text{IOUpred} = \Pr(\text{Class}_i) \ast \text{IOUpred}$$

which gives us class-specific confidence scores for each box. These scores encode both the probability of that class appearing in the box and how well the predicted box fits the object

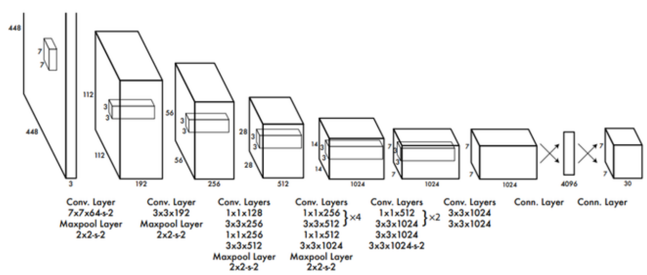


We implement this model as a convolutional neural net- work and evaluate it on the PASCAL VOC detection dataset [9]. The initial convolutional layers of the network extract features from the image while the fully connected layers predict the output probabilities and coordinates.

Our network architecture is inspired by the GoogLeNet model for image classification [33]. Our network has 24 convolutional layers followed by 2 fully connected layers. Instead of the inception modules used by GoogLeNet, we simply use 1 × 1 reduction layers followed by 3 × 3 convo- lutional layers.

We also train a fast version of YOLO designed to push the boundaries of fast object detection. Fast YOLO uses aneural network with fewer convolutional layers (9 insteadof 24) and fewer filters in those layers. Other than the sizeof the network, all training and testing parameters are thesamebetweenYOLOandFastYOLO

ALGORITHM



The Architecture. Our detection network has 24 convolutional layers followed by 2 fully connected layers. Alternating 1×1

convolutional layers reduce the features space from preceding layers. We pretrain the convolutional layers on the ImageNet classification task at half the resolution (224×224 input image) and then double the resolution for detection.

Real-Time Detectors	Train	mAP	FPS
100Hz DPM [30]	2007	16.0	100
30Hz DPM [30]	2007	26.1	30
Fast YOLO	2007+2012	52.7	**155**
YOLO	2007+2012	**63.4**	45
Less Than Real-Time			
Fastest DPM [37]	2007	30.4	15
R-CNN Minus R [20]	2007	53.5	6
Fast R-CNN [14]	2007+2012	70.0	0.5
Faster R-CNN VGG-16[27]	2007+2012	73.2	7
Faster R-CNN ZF [27]	2007+2012	62.1	18
YOLO VGG-16	2007+2012	66.4	21

MAP" stands for Mean Average Precision, and "FPS" stands for Frames Per Second

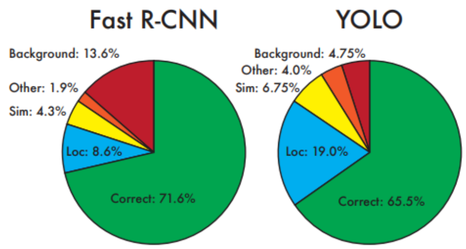


Figure 4: Error Analysis: Fast R-CNN vs. YOLO These charts show the percentage of localization and background errors in the top N detections for various categories (N = # objects in that category).

VOC 2012 test	mAP	aero	bike	bird	boat	bottle	bus	car	cat	chair	cow	table	dog	horse	mbike	person	plant	sheep	sofa	train	tv
MR_CNN_MORE_DATA [11]	**73.9**	**85.5**	**82.9**	**76.6**	**57.8**	**62.7**	**79.4**	77.2	86.6	**55.0**	**79.1**	**62.2**	87.0	**83.4**	**84.7**	78.9	45.3	73.4	65.8	80.3	74.0
HyperNet_VGG	71.4	84.2	78.5	73.6	55.6	53.7	78.7	**79.8**	87.7	49.6	74.9	52.1	86.0	81.7	83.3	**81.8**	**48.6**	**73.5**	59.4	79.9	65.7
HyperNet_SP	71.3	84.1	78.3	73.3	55.5	53.6	78.6	79.6	87.5	49.5	74.9	52.1	85.6	81.6	83.2	81.6	48.4	73.2	59.3	79.7	65.6
Fast R-CNN + YOLO	70.7	83.4	78.5	73.5	55.8	43.4	79.1	73.1	**89.4**	49.4	75.5	57.0	**87.5**	80.9	81.0	74.7	41.8	71.5	68.5	**82.1**	67.2
MR_CNN_S_CNN [11]	70.7	85.0	79.6	71.5	55.3	57.7	76.0	73.9	84.6	50.5	74.3	61.7	85.5	79.9	81.7	76.4	41.0	69.0	61.2	77.7	72.1
Faster R-CNN [27]	70.4	84.9	79.8	74.3	53.9	49.8	77.5	75.9	88.5	45.6	77.1	55.3	86.9	81.7	80.9	79.6	40.1	72.6	60.9	81.2	61.5
DEEP_ENS_COCO	70.1	84.0	79.4	71.6	51.9	51.1	74.1	72.1	88.6	48.3	73.4	57.8	86.1	80.0	80.7	70.4	46.6	69.6	**68.8**	75.9	71.4
NoC [28]	68.8	82.8	79.0	71.6	52.3	53.7	74.1	69.0	84.9	46.9	74.3	53.1	85.0	81.3	79.5	72.2	38.9	72.4	59.5	76.7	68.1
Fast R-CNN [14]	68.4	82.3	78.4	70.8	52.3	38.7	77.8	71.6	89.3	44.2	73.0	55.0	**87.5**	80.5	80.8	72.0	35.1	68.3	65.7	80.4	64.2
UMICH_FGS_STRUCT	66.4	82.9	76.1	64.1	44.6	49.4	70.3	71.2	84.6	42.7	68.6	55.8	82.7	77.1	79.9	68.7	41.4	69.0	60.0	72.0	66.2
NUS_NIN_C2000 [7]	63.8	80.2	73.8	61.9	43.7	43.0	70.3	67.6	80.7	41.9	69.7	51.7	78.2	75.2	76.9	65.1	38.6	68.3	58.0	68.7	63.3
BabyLearning [7]	63.2	78.0	74.2	61.3	45.7	42.7	68.2	66.8	80.2	40.6	70.0	49.8	79.0	74.5	77.9	64.0	35.3	67.9	55.7	68.7	62.6
NUS_NIN	62.4	77.9	73.1	62.6	39.5	43.3	69.1	66.4	78.9	39.1	68.1	50.0	77.2	71.3	76.1	64.7	38.4	66.9	56.2	66.9	62.7
R-CNN VGG BB [13]	62.4	79.6	72.7	61.9	41.2	41.9	65.9	66.4	84.6	38.5	67.2	46.7	82.0	74.8	76.0	65.2	35.6	65.4	54.2	67.4	60.3
R-CNN VGG [13]	59.2	76.8	70.9	56.6	37.5	36.9	62.9	63.6	81.1	35.7	64.3	43.9	80.4	71.6	74.0	60.0	30.8	63.4	52.0	63.5	58.7
YOLO	57.9	77.0	67.2	57.7	38.3	22.7	68.3	55.9	81.4	36.2	60.8	48.5	77.2	72.3	71.3	63.5	28.9	52.2	54.8	73.9	50.8
Feature Edit [32]	56.3	74.6	69.1	54.4	39.1	33.1	65.2	62.7	69.7	30.8	56.0	44.6	70.0	64.4	71.1	60.2	33.3	61.3	46.4	61.7	57.8
R-CNN BB [13]	53.3	71.8	65.8	52.0	34.1	32.6	59.6	60.0	69.8	27.6	52.0	41.7	69.6	61.3	68.3	57.8	29.6	57.8	40.9	59.3	54.1
SDS [16]	50.7	69.7	58.4	48.5	28.3	28.8	61.3	57.5	70.8	24.1	50.7	35.9	64.9	59.1	65.8	57.1	26.0	58.8	38.6	58.9	50.7
R-CNN [13]	49.6	68.1	63.8	46.1	29.4	27.9	56.6	57.0	65.9	26.5	48.7	39.5	66.2	57.3	65.4	53.2	26.2	54.5	38.1	50.6	51.6

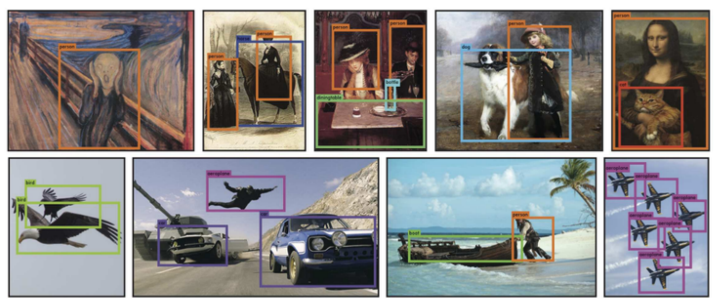
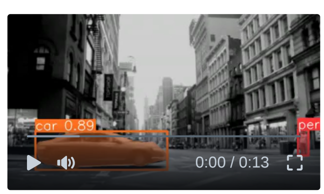


Figure 6: Qualitative Results. YOLO running on sample artwork and natural images from the internet. It is mostly accurate although it does think one person is an airplane.

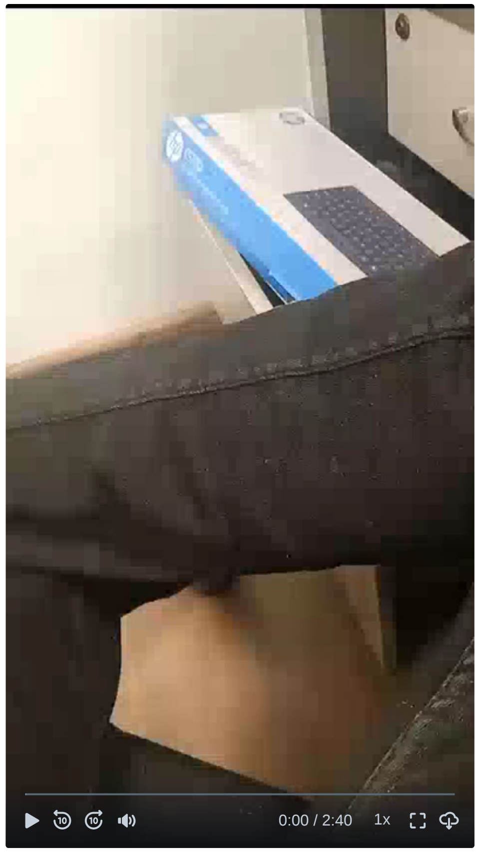
Some of the processed videos At leos lab

Segmentation

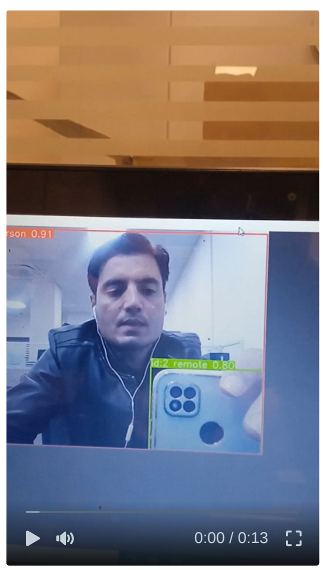


Tracking

IMPLEMENTED ON RADXA



FURTHER PROGRESS



Phases	DATE	TASK ASSIGNED
PHASE 0	18/1/24 till -	Understanding the algorithm in detail

PROGRESS

This is the link of paper I have decided to do my comprehensive studies as per the official phase plan approved by admin.

link

https://www.cv-foundation.org/openaccess/content_cvpr_2016/papers/Redmon_You_Only_Look_CVPR_2016_paper.pdf

Market Research

Project: DNSS

Prepared by: JTM ABDULLAH KHALID

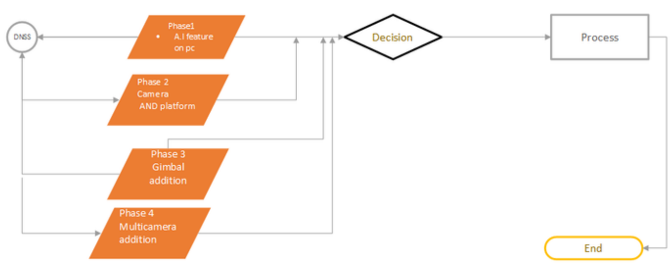
Phase 1: Research and Conceptualization - Conduct extensive market research to identify emerging trends and user needs. Define a set of artificial features, such as image recognition, augmented reality, and intelligent scene analysis, that align with consumer demands.

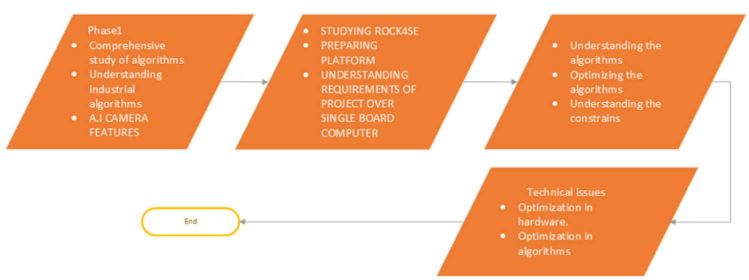
Phase 2: Design and Prototyping - Develop a detailed design incorporating the chosen artificial features. Create a functional prototype to test the integration of these features, ensuring seamless performance and user-friendly interfaces.

Phase 3: Testing and Optimization - Rigorous testing to evaluate the prototype's reliability, accuracy, and overall performance. Collect user feedback to identify areas for improvement, and optimize the camera's algorithms and software for enhanced functionality and responsiveness.

Phase 4: Manufacturing and Assembly - Scale up production based on successful prototype testing. Collaborate with manufacturing partners to ensure mass production meets quality standards while maintaining cost-effectiveness.

PHASE 1	**A.I features on pc**
PHASE 2	**Camera + platform**
PHASE 3	**Gimbal addition**
PHASE 4	**Multicamera addition**





Phases	DATE	TASK ASSIGNED

PHASE 0	18/1/24 till -	Understanding the algorithm in detail

PROGRESS

This is the link of paper I have decided to do my comprehensive studies as per the official phase plan approved by admin.

link

https://www.cv-foundation.org/openaccess/content_cvpr_2016/papers/Redmon_You_Only_Look_CVPR_2016_paper.pdf

For optimized performance when implementing YOLOv1, you can tailor the requirements to ensure efficient training and inference. Here's an expanded definition of the key requirements in tabular form:

Category	Requirements	Explanation
Hardware		
GPU	NVIDIA GPU (e.g., GeForce GTX or Quadro series)	A powerful GPU with CUDA support for accelerated training and inference.
CUDA-enabled GPU	Compute Unified Device Architecture	A GPU architecture designed by NVIDIA to enhance parallel processing. Required for GPU acceleration.
Compute Capability	Compute Capability 3.0 or higher	A specific version of the CUDA architecture. Check compatibility with the chosen deep learning framework.
VRAM (Video RAM)	8GB or higher recommended	Sufficient video memory for handling large batches and deep neural networks during training and inference.
Software		
Operating System	Linux (e.g., Ubuntu) or Windows	Linux is commonly used for deep learning tasks, but YOLO can also be implemented on Windows.
CUDA Toolkit	CUDA Toolkit 7.5 or later	A parallel computing platform and API model created by NVIDIA. Required for GPU acceleration with YOLO.
cuDNN Library	cuDNN v5 or later	NVIDIA's deep neural network library. Optimizes deep neural network computations.
Python	Python 3.x	YOLO is typically implemented using Python. Python 3.x is commonly used.
Deep Learning Framework	Darknet, TensorFlow, or PyTorch	Darknet is the original framework for YOLO. TensorFlow and PyTorch have versions adapted for YOLO.
YOLO-Specific		
Darknet Framework	Darknet v1.0 or later	The reference implementation for YOLO. Includes the YOLOv1 architecture and training pipeline.
Configuration Files	yolo.cfg, obj.data, obj.names	Modify configuration files for model architecture, training parameters, and class names.

Pre-trained Weights	yolov1.weights or transfer learning weights	Optionally use pre-trained weights for initialization, especially if employing transfer learning.
Dataset	Annotated dataset with bounding boxes	Prepare a dataset with annotated images and bounding boxes for the objects of interest.
Data Preprocessing	Augmentation, normalization, resizing	Implement data augmentation and preprocessing to enhance model generalization.
Training Parameters		
Batch Size	Adjust based on GPU memory (e.g., 64, 128)	Set an appropriate batch size for efficient GPU memory usage. Smaller batches may be needed for limited VRAM.
Learning Rate	Tune based on convergence (e.g., 0.001 to 0.0001)	Optimize the learning rate for effective model convergence.
Number of Epochs	Based on convergence and dataset size	Define the number of training epochs for sufficient convergence.
Input Image Size	Trade-off between speed and accuracy (e.g., 416)	Adjust the input image size to balance processing speed and detection accuracy.
Anchor Boxes	Extracted from the dataset or predefined	Define anchor boxes to assist in object localization.