# GAMMA

1. Intro and Dataset

GAMMA(Glaucoma grading from Multi-Modality imAges):

<https://aistudio.baidu.com/aistudio/competition/detail/90>

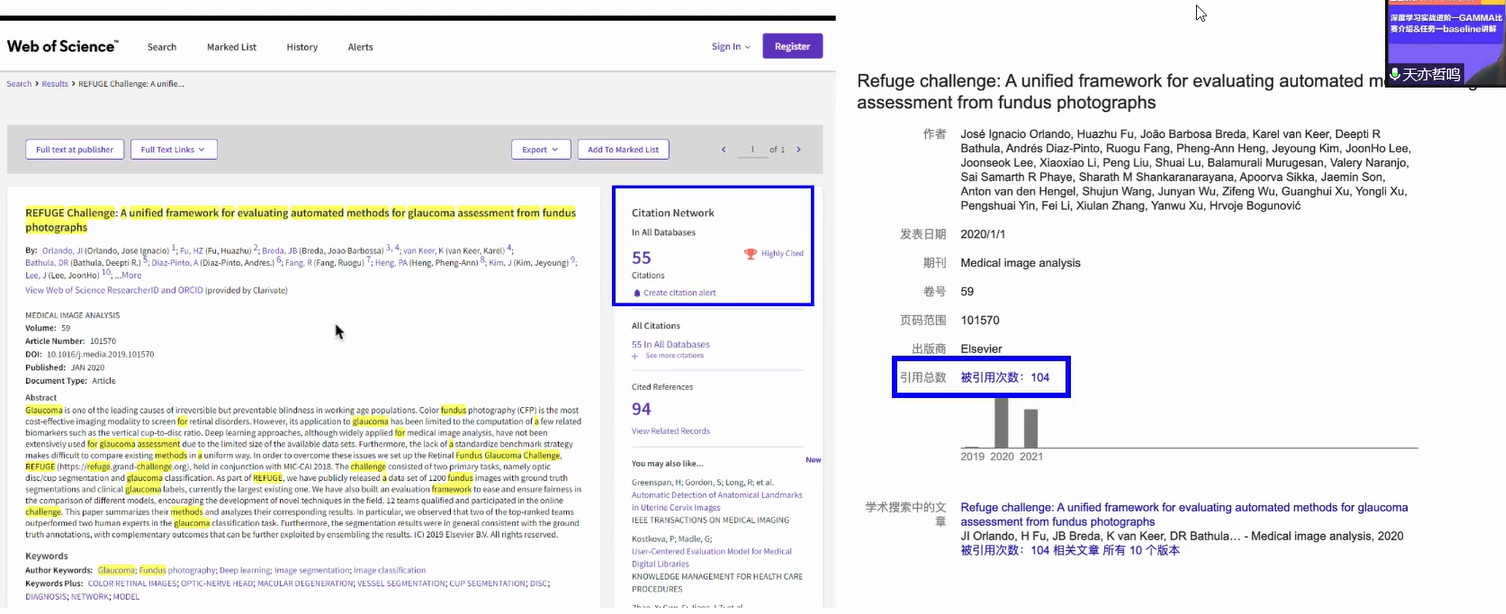
MICCAI (International Conference on Medical Image Computing and Computer Assisted Intervention): <https://miccai2021.org/en/>

OMIA(Ophthalmic Medical Image Analysis): <https://omia.grand-challenge.org/challenge>

International Council of Ophthalmology (ICO): <http://www.icoph.org/>, <http://ies.ijo.cn/gjykcn/ch/index.aspx>

Former Challenges and Papers







比赛成绩计算：

Scoretotal =0.3∗Scorepreliminary+0.7∗Scorefinal

​ 

规则：不能使用外部眼底标注数据，不能多模态青光眼分级任务中不能只用一模态数据

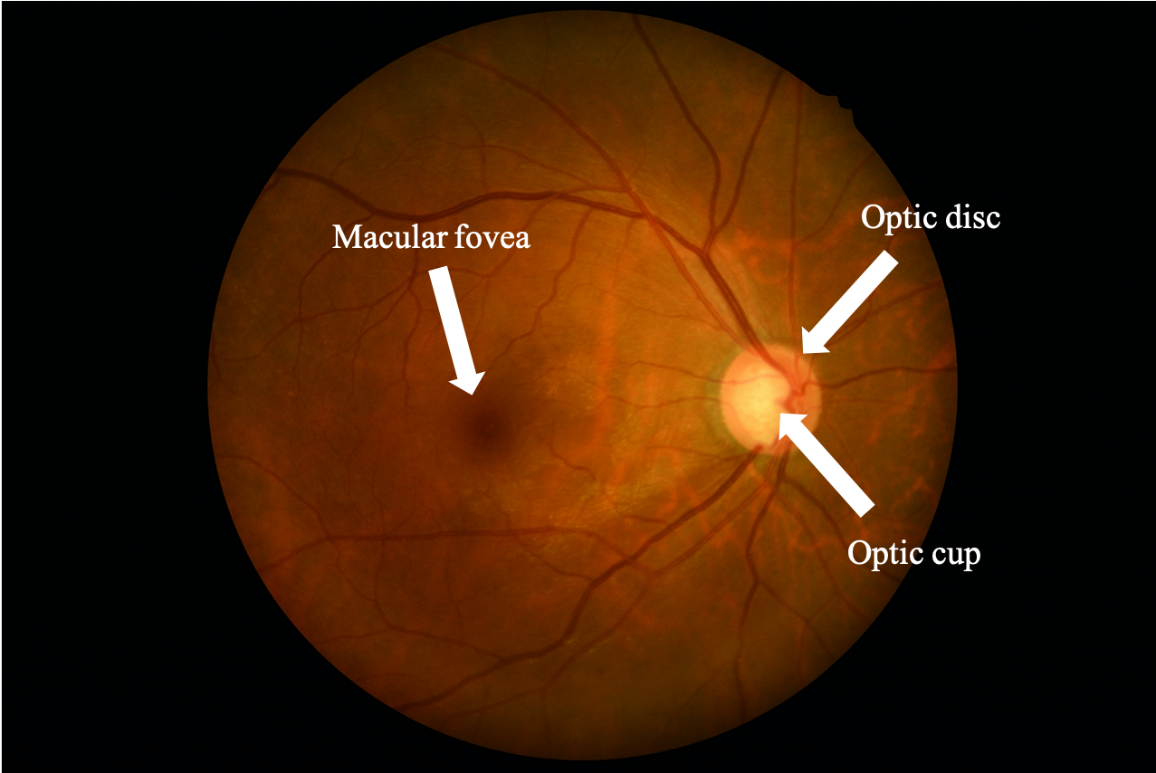
训练样本100，预赛样本100，决赛样本100

目的：针对医学图像的小样本训练

视野检查： MD30-2，早中晚期分级标准：MD>-6dB，-6dB≥MD≥-12dB， MD<-12dB

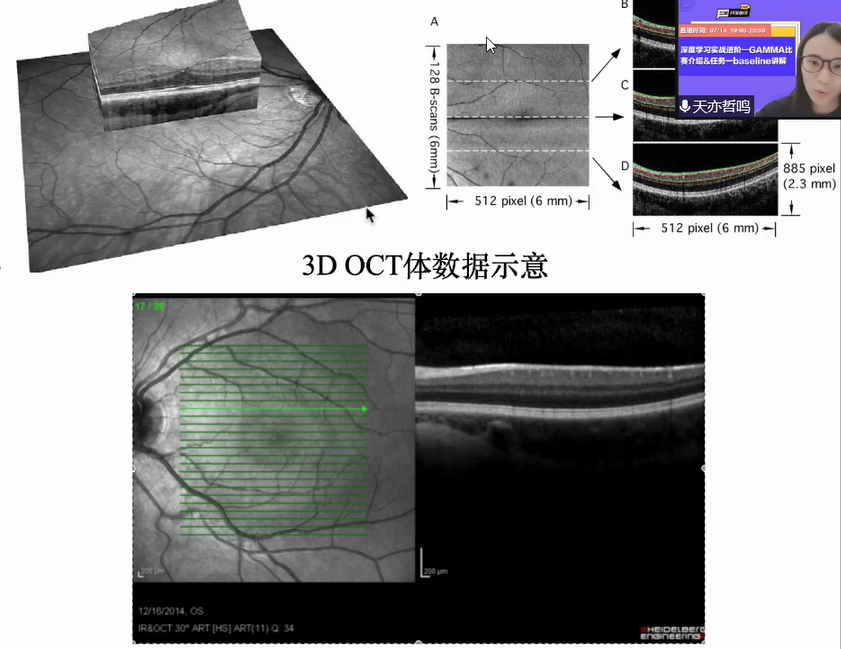
视神经垂直杯盘比：早≥0.5，中晚≥0.7

2D眼底彩照: Optic cup is within Optic disc, fundus，由不同机型扫描，100张



尺寸(H, W, C)：(1934, 1956, 3), (2000, 2992, 3)

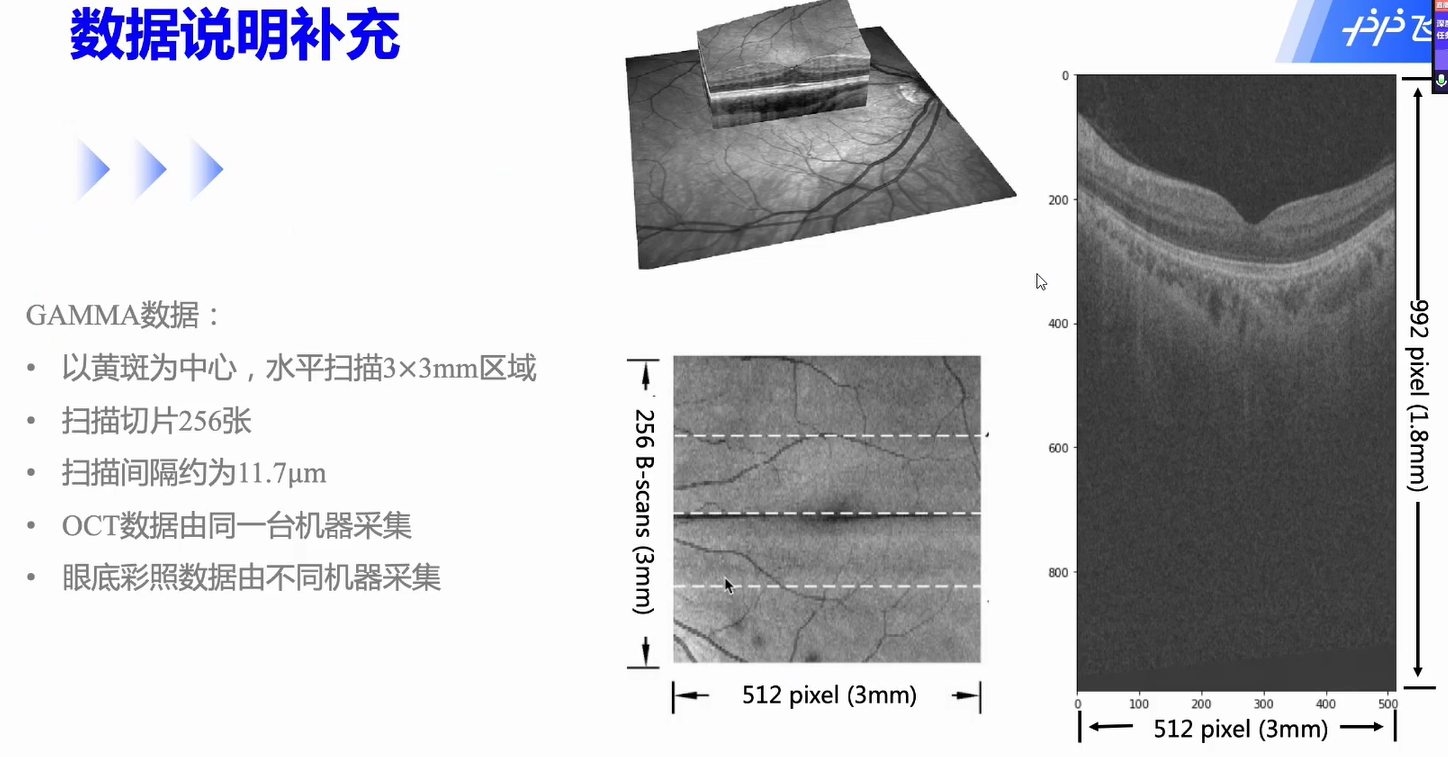
Spectral Domain OCT, SD-OCT, 3D眼底OCT数据



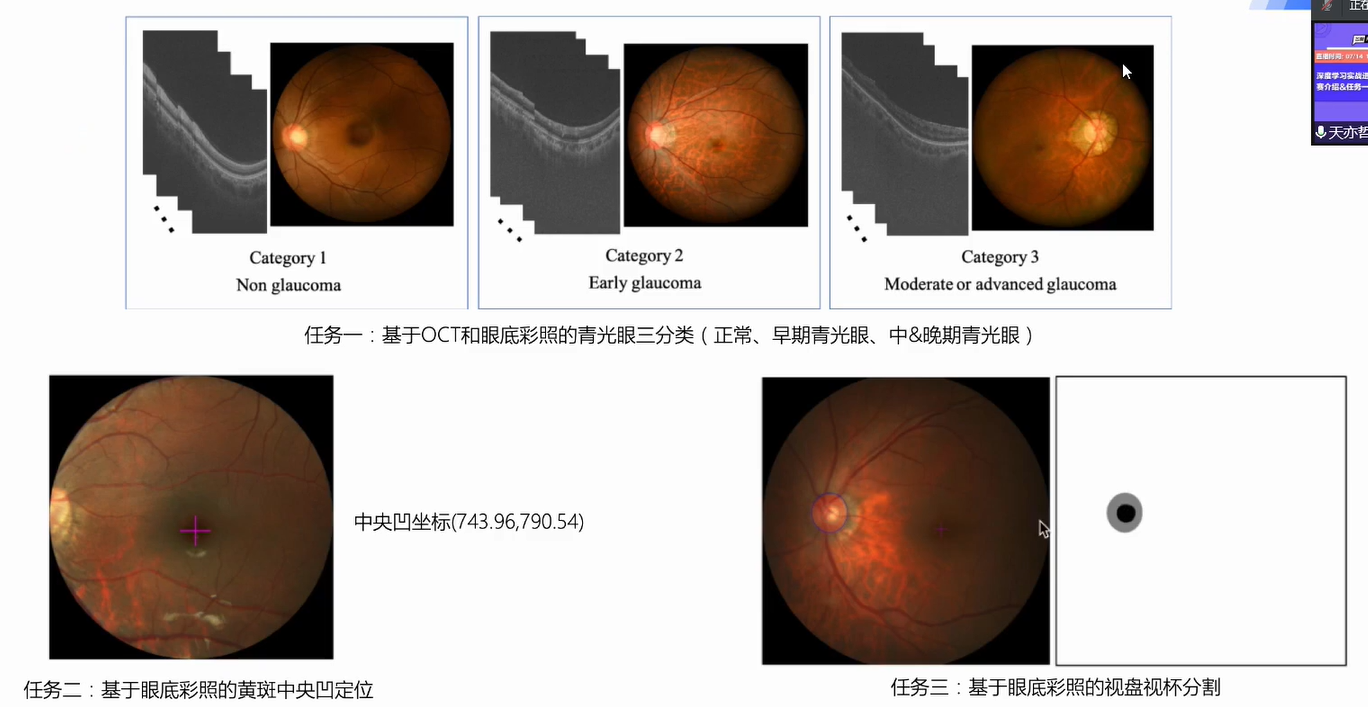
OCT切片以黄斑为中心扫描，3mmx3mm，不一定覆盖到optic disc. 青光眼视神经纤维层、神经细胞复合体层会变薄

OCT扫描间距 11.7μm（机器精度10μm，有误差）

OCT不同个体数据扫描参数有差别



**Tasks:** 1) glaucoma grading, 2) macular fovea localization, 3) optic disc and cup segmentation.



1. Task 1 Glaucoma Grading Baseline

**Processing Training Dataset**



input: N×3×512×512，N×256×512×512

**Code**

Model: concatenated ResNet34

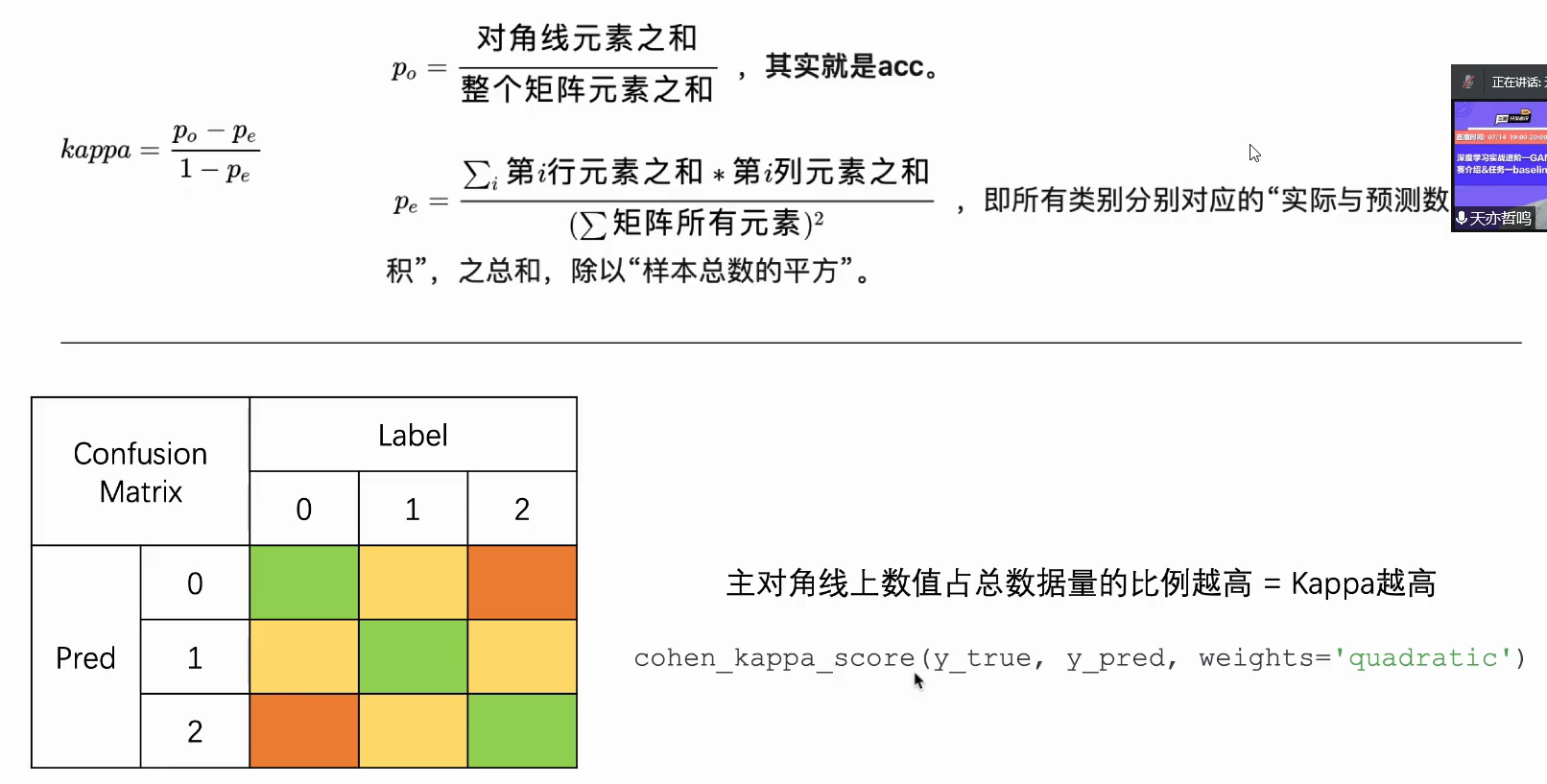
paddle里的交叉熵不需要激活

32G V100 batchsize=6 但瓶颈在dataloader (CPU

**CE Loss**

**Testing Metrics**

use **kappa** score instead of **accuracy** score



baseline **kappa** = 0.73，1400 steps

**Tricks of Improvement：**

different ways to ensemble

local info of optic disc: attention, segment, crop

1. Task2 Macular Fovea Localization Baseline

**Image Processing**

without data augmentation

coordinates normalization

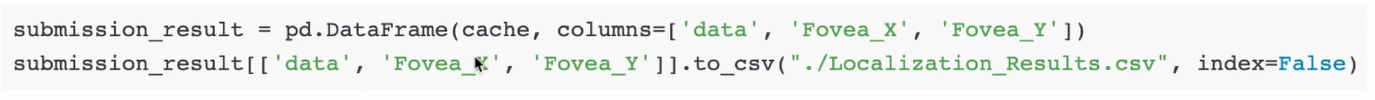
resize 256×256

**Model:** ResNet50 + FC layer + regression to (x,y)

**Loss:** 0.5MSE Loss + 0.5 Euclidean Distance (selectable: L1 L2 Loss)

**Test Score:**



**Save and evaluate Test results:** 

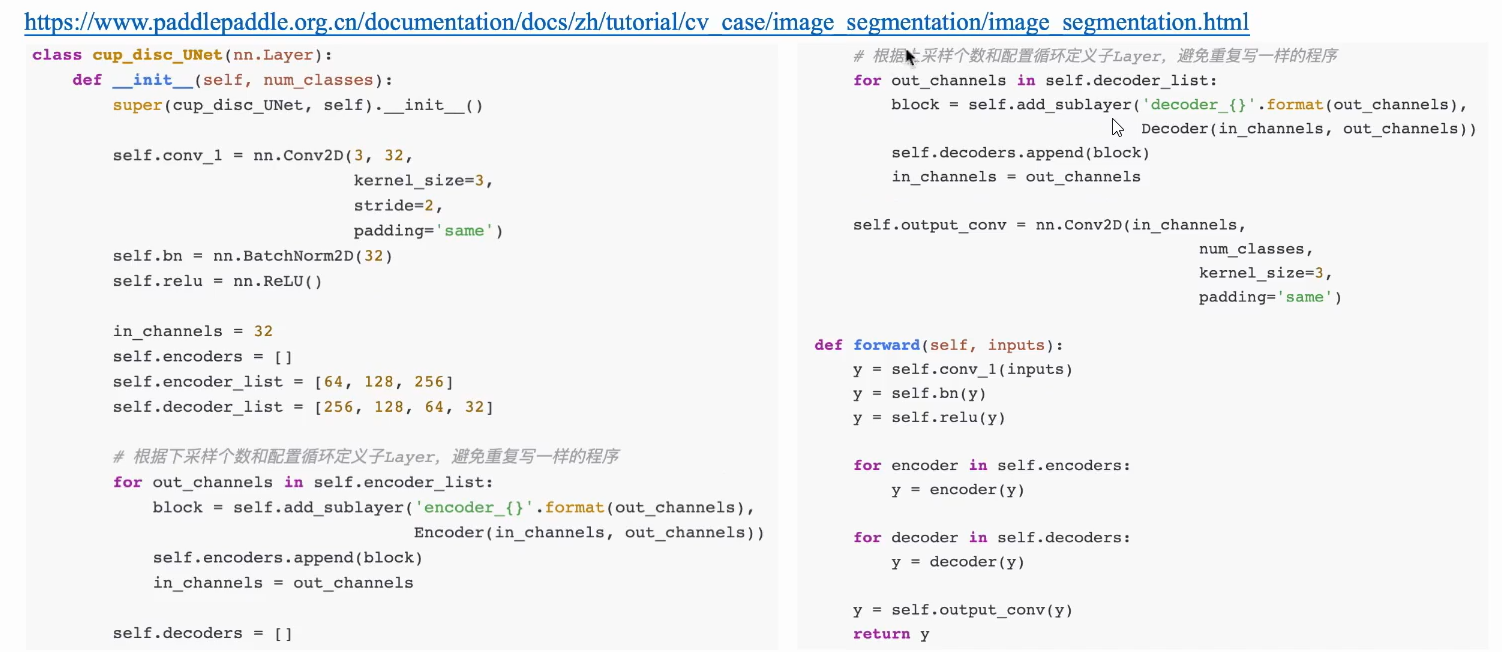
**Tricks to Improve the performance:**

data augmentation

1. Task3 Optic Disc and Cup Segmentation Baseline

**per-pixel classification:** class 0 - optic cup (pixel value =0 ), class 1 - optic disc(pixel value = 0, 128), class 2 - background (pixel value =255)

**Model:** Unet

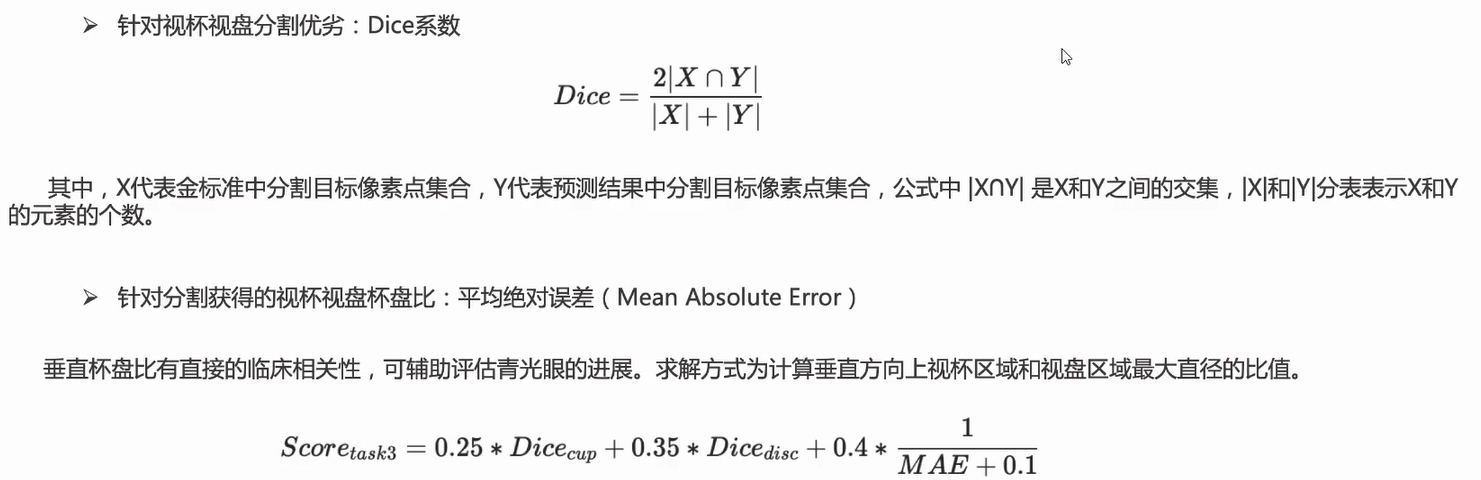


**Loss:** CE Loss (Cross Entropy)

**Test Metrics:**

Segmentation: Dice

cup-disc ratio: Mean Absolute Error



**Baseline Results: Dice 0.79+**

**Improvements:**

UNet + different scales, attention,

# Develop a Model on Paddle

*https://www.paddlepaddle.org.cn/documentation/docs/zh/guides/02\_paddle2.0\_develop/index\_cn.html*

## Definition and Dataloader of Dataset

## Data Preprocessing

**Data augmentations**: random rotation, horizontal flip, random crop/loop

print('数据处理方法：', paddle.vision.transforms.\_\_all\_\_)

from paddle.vision.transforms import Compose, Resize

*self.transform = Compose([Resize(size=32)* *, RandomHorizontalFlip()])*

## Model Ensemble

**Attention**

*self.transformer1= paddle.nn.Transformer()*

**2 methods of Model Ensembling on Paddle**

Sequential

Subclass

**2 ways to call embed network/layer on Paddle**

*paddle.nn.Conv2D*

*paddle.vision.models.resnet34(pretrained=True, num\_classes=0) # remove final fc*

**see all of the embed architectures and layers**

*print('飞桨框架内置模型：', paddle.vision.models.\_\_all\_\_)*

*print(paddle.nn.\_\_all\_\_)*

## Training and Inferencing

**2 approaches to Training and 3 ways to Predicting  
 Advanced APIs**：*Model.prepare()、Model.fit()、Model.evaluate()、Model.predict()*

define train() and val() by **Basic APIs**: *model.train(), model.eval(), optim.step(), loss.backward()*

**模型的部署预测paddle.inference*:*** *paddle.jit.save, inference.Config(), inference.* *create\_predictor()*

## Saving and Loading of Models

Saving and Loading of **basic APIs**

**训练调优**场景的模型&参数保存载入

**动态图参数保存载入**

***paddle.save****(model.state\_dict(), “PATH/* A.***pdiparams****”)*

*paddle.save(Optimizer.state\_dict(),“PATH/B.****pdopt****”)*

*paddle.save(emb.weight) – pdtensor*

***paddle.load***

***model.set\_state\_dict****(model\_state\_dict)*

***optimizer.set\_state\_dict****(opt\_state\_dict)*

**静态图模型&参数保存载入**

***paddle.save(****prog.state\_dict(),**"temp/model.****pdparams****"****)***

***paddle.save****(prog, "temp/model****.pdmode****l")*

*prog =* ***paddle.load****("temp/model****.pdmodel****")*

*state\_dict =* ***paddle.load****("temp/model****.pdparams****")*

***prog.set\_state\_dict(state\_dict)***

**训练部署**场景的模型&参数保存载入

**动态图模型&参数保存载入（训练推理**）：动转静训练 + 模型&参数保存；动态图训练 + 模型&参数保存

***paddle.jit.save****(layer=,“PATH/model”)***模型部署** *- pdmodel, pdiparams.info, pdiparams*

*@paddle.jit.to\_static*

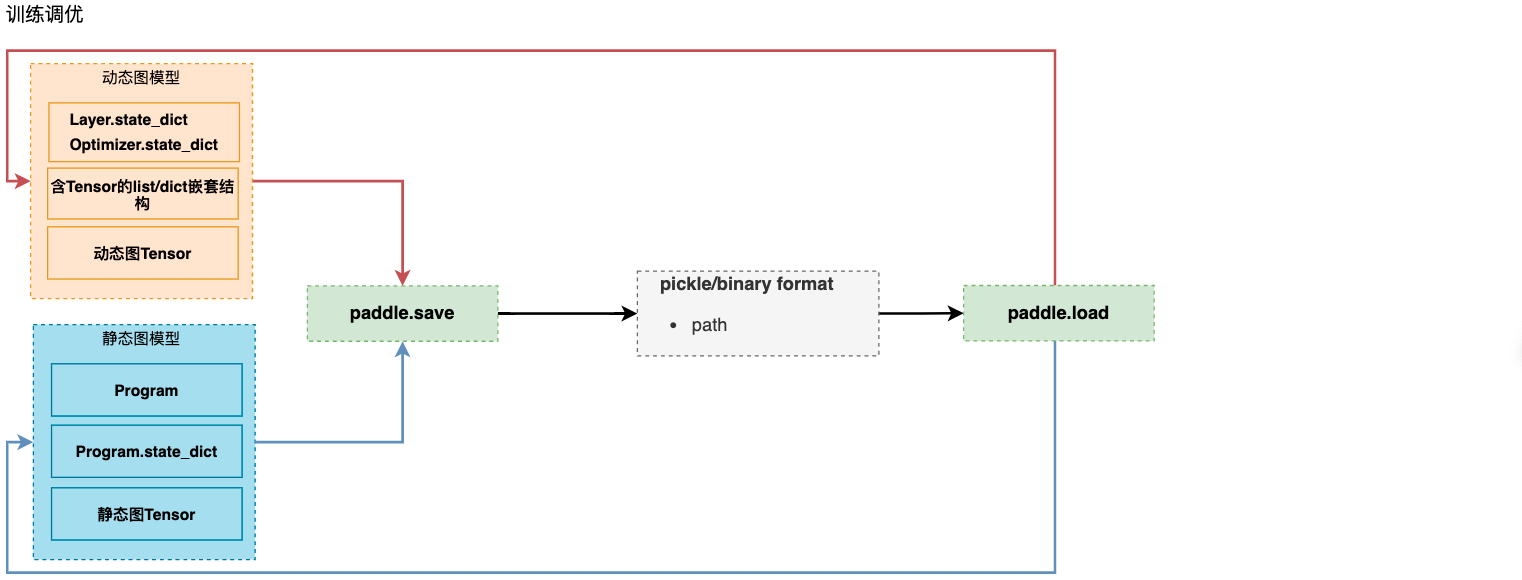
***paddle.jit.load***

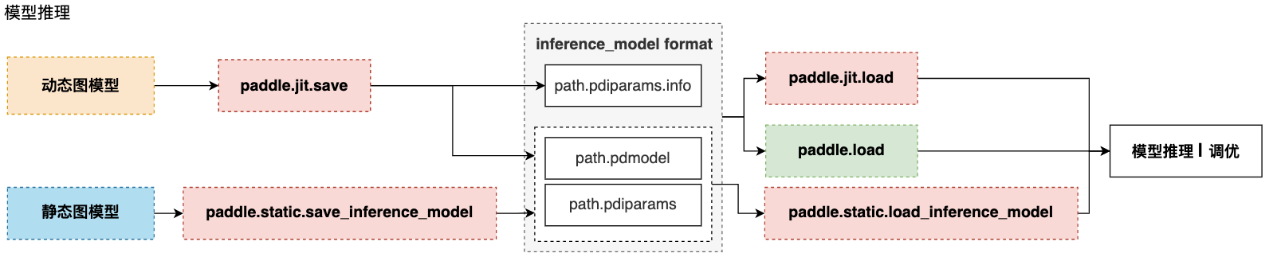
**静态图模型&参数保存载入（推理部署）**

***paddle.static.save\_inference\_model***

***paddle.static.load\_inference\_model***

**各接口关系**





Saving and Loading of **Advanced APIs**

**paddle.Model.fit (训练接口，同时带有参数保存的功能)**

**paddle.Model.save**

**paddle.Model.load**

## Distributed Training

[*https://www.paddlepaddle.org.cn/documentation/docs/zh/guides/02\_paddle2.0\_develop/06\_device\_cn.html*](https://www.paddlepaddle.org.cn/documentation/docs/zh/guides/02_paddle2.0_develop/06_device_cn.html)

*https://www.paddlepaddle.org.cn/documentation/docs/zh/guides/06\_distributed\_training/index\_cn.html*

**Distributed Training**

**paddle.distributed.launch以文件为单位:** advanced API & Basic API

**paddle.distributed.spawn**(dist.spawn(train, args=(True,), nprocs=2), gpus='4,5')

## Customized Loss, Metrics and Callback

**ModelCheckpoint**回调函数，可以在”fit”训练模型时自动存储**每轮**训练得到的模型。

## Visualization

**Summary**

*paddle.summary(lenet, (64, 1, 28, 28))*

# Visual DL kit

**Installation:** *pip install --upgrade --pre visualdl*

**LogWriter:**

*with LogWriter(logdir="./log/scalar\_test/train") as writer:*

***writer.add\_scalar****(tag="acc", step=1, value=0.5678)\*

**Launch Visual DL:**

Shell: *visualdl --logdir <dir\_1, dir\_2, ... , dir\_n> --host <host> --port <port> --cache-timeout <cache\_timeout> --language <language> --public-path <public\_path> --api-only*

**Py Script:** *visualdl.server.app.run(logdir,*

*host="127.0.0.1",*

*port=8080,*

*cache\_timeout=20,*

*language=None,*

*public\_path=None,*

*api\_only=False,*

*open\_browser=False)*

**LogReader**

**Scalar:** 动态展示，多实验对比

**Image:** real-time display

**Audio**

**Graph:** visualize the architecture. support a host of DL framework

**Histogram:** Offset mode, Overlay mode

**PR Curve:** *add\_pr\_curve(tag, labels, predictions, step=None, num\_thresholds=10)*

**High Dimensional:** (Dimensionality reduction) t-SNE, PCA

# From Dynamic Graphs to Static Graphs

[*https://www.paddlepaddle.org.cn/documentation/docs/zh/guides/04\_dygraph\_to\_static/index\_cn.html*](https://www.paddlepaddle.org.cn/documentation/docs/zh/guides/04_dygraph_to_static/index_cn.html)

**Dynamic Graphs:** 适合**训练调优** 易用的接口，Python风格的编程体验，友好的**debug**交互机制等。在动态图模式下，**代码是按照我们编写的顺序依次执行**。

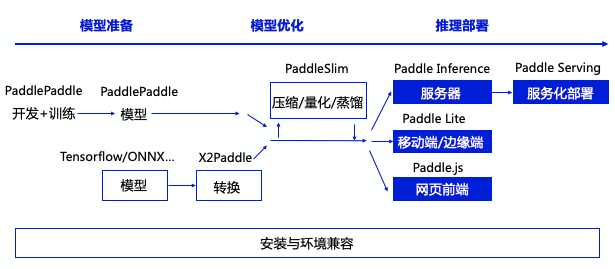
在**工业界**的许多**部署**场景中（如大型推荐系统、移动端）都倾向于直接使用**C++**来提速

**Static Graph:** 在**部署**方面更具有性能的优势。静态图程序在编译执行时，**先搭建模型的神经网络结构**，然后再对神经网络执行**计算**操作。预先搭建好的神经网络可以**脱离Python依赖，在C++端被重新解析执行**，而且拥有整体网络结构也能进行一些网络结构的优化

# Deploy Inference Model

<https://www.paddlepaddle.org.cn/documentation/docs/zh/guides/05_inference_deployment/index_cn.html>

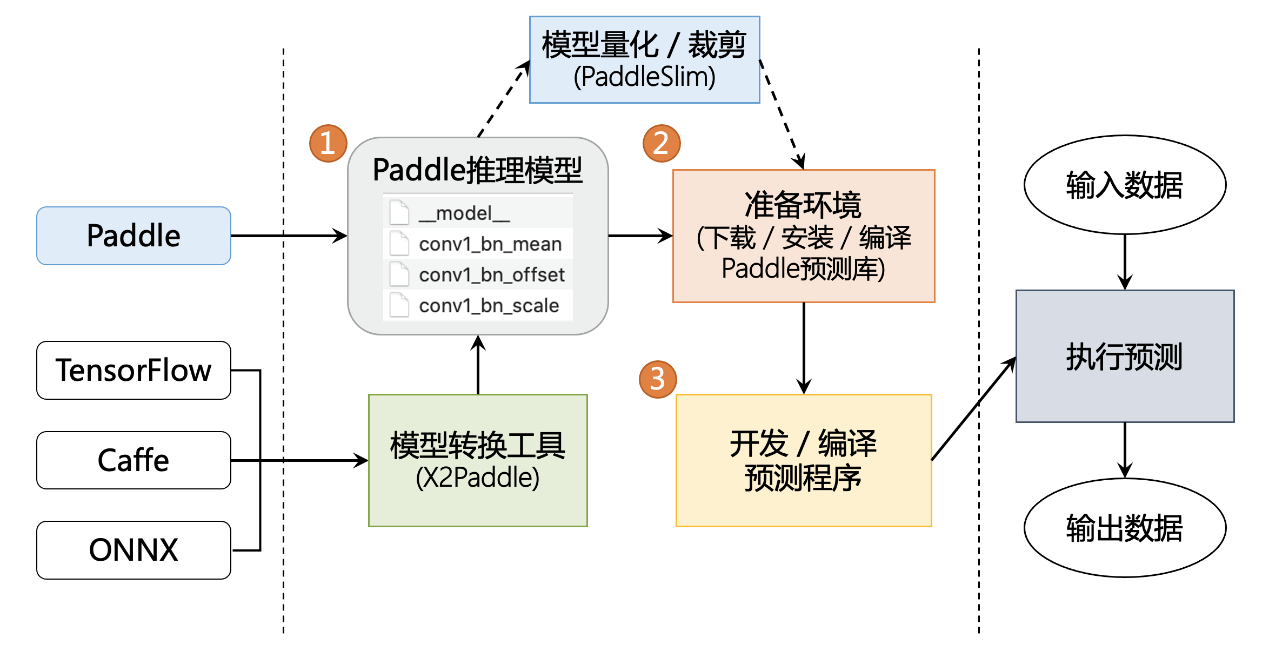
**Workflow:**



PaddlePaddle, X2Paddle

## Paddle Inference

主流软硬件环境、主流和国产操作系统、多语言接口兼容适配



## Paddle Serving, Paddle Lite, Paddle.js

# Model Compression- PaddleSim

[*https://www.paddlepaddle.org.cn/documentation/docs/zh/guides/05\_inference\_deployment/paddleslim/paddle\_slim\_cn.html*](https://www.paddlepaddle.org.cn/documentation/docs/zh/guides/05_inference_deployment/paddleslim/paddle_slim_cn.html)

Pruning

Fixed Point Quantization

Knowledge Distillation

Neural Architecture Search (NAS)

# Write Customized Operator and Contribute to Paddle

# Export from and Import to Paddle

## Exporting Models to ONNX (Open Neural Network Exchange)

[*https://www.paddlepaddle.org.cn/documentation/docs/zh/guides/02\_paddle2.0\_develop/09\_model\_to\_onnx\_cn.html*](https://www.paddlepaddle.org.cn/documentation/docs/zh/guides/02_paddle2.0_develop/09_model_to_onnx_cn.html)

ONNX can be inferenced by other frameworks, such as OpenVINO, ONNX Runtime.

**Paddle1 静态图组; Paddle2.0开始主推动态图组网**

Process of exporting Paddle to ONNX:

***paddle2onnx***

*transfer dynamic graphs to static graphs*

***paddle.onnx.export()***

*onnx\_model =* ***onnx.load****(onnx\_file)*

***onnx.checker.check\_model(****onnx\_model****) #*验证算子是否符合对应版本的协议，及网络结构是否完整**

***onnxruntime.InferenceSession****(onnx\_file) # inference*

## API mapping table of Paddle 1.8 and Paddle 2.0

## X2Paddle

[*https://www.paddlepaddle.org.cn/documentation/docs/zh/guides/08\_api\_mapping/pytorch\_api\_mapping\_cn.html*](https://www.paddlepaddle.org.cn/documentation/docs/zh/guides/08_api_mapping/pytorch_api_mapping_cn.html)

* **预测模型转换**
  + 支持Caffe/TensorFlow/ONNX/PyTorch的模型一键转为飞桨的预测模型，并使用PaddleInference/PaddleLite进行CPU/GPU/Arm等设备的部署
* **PyTorch训练项目转换**
  + 支持PyTorch项目**Python代码（包括训练、预测）**一键转为基于飞桨框架的项目代码，帮助开发者快速迁移项目，并可享受AIStudio平台对于飞桨框架提供的海量免费计算资源
* **API映射文档**
  + 详细的**API文档对比分析**，帮助开发者快速从**PyTorch**框架的使用迁移至飞桨框架的使用，大大降低学习成本

详细的项目信息与使用方法参考X2Paddle在Github上的开源项目: [*https://github.com/PaddlePaddle/X2Paddle*](https://github.com/PaddlePaddle/X2Paddle)

# Applications

visit the link below to read applications in diverse domains:

<https://www.paddlepaddle.org.cn/documentation/docs/zh/tutorial/index_cn.html>

CV

NLP

RL

Time Series Data (TSD)

Recommendation Algorithm

# Improvements with Code (比赛调优方法)

**Data augmentations –** fundus images**:** random rotation, horizontal flip, random crop/loop

print('数据处理方法：', paddle.vision.transforms.\_\_all\_\_)

from paddle.vision.transforms import Compose, Resize

self.transform = Compose([Resize(size=32) , RandomHorizontalFlip()])

**断点训练 增量训练 transfer learning √**

**checkpoint**

**pdmodel, pdparams, pdopt**

*paddle.save(Optimizer.state\_dict(),“PATH/B.pdopt”)*

*optimizer.set\_state\_dict(opt\_state\_dict)*

**model ensemble**

**Visualization**

**paddle.summary(lenet, (64, 1, 28, 28))**

**Improvements of 3 Tasks**

**Task1**

1. **Data Pre-processing – OCT images**

crop 2/3 in 992-pixel side instead of CenterCrop

resize to 256 \* 256

threshold segmentation, set pixel(grey value<90) = 0

1. resnet50, resnet101, resnet 152
2. attention
3. combine the results of task3
4. add those 2 line in Task1

images\_file.**sort()**

**random\_state = 42**

保证训练时划分的验证集一致

1. **task2黄斑定位与task3 optic disk/cup分割助力task1**
   1. 叠加2个通道，5通道可行吗？

**Task2**

1. Data Augmentation√
   1. Centre\_Crop and Resize，
   2. RHorizonFlip，
   3. RVerticalFlip，
   4. Rrotation # the range of rotation is (-60, 60)
2. ImageNet Pretrained =True √
3. global, local, micro-local （根据GT点定位一个小范围）**多尺度信息学习**
   1. 在 global和local确定大致范围后，**裁剪出一个小区域(128×128)进行训练（2 Step Network由粗到细）**，**注意这时的映射关系不能用中心裁剪映射：**resnet50, resnet34
      1. bad performance: resnet34, colour jitter(0.2, 0.2, 0.2, 0.1), learning rate sgd, alpha =0.5
   2. **应该有其他金字塔型网络能达到类似的效果**
   3. pretrain micro-local info, because it’s not accessible when testing
4. New network
5. 可转化为**目标定位**任务训练（采用**高斯**等**得到一个范围**）- **参考GR-ConvNet得到抓取中心点**
6. 转化为分割任务：分割→取中心点or micro-local小范围回归得到中心凹 **，由粗到细**
   1. 分割任务的**误差为像素级1/image\_size**
7. **视杯视盘分割与黄斑定位结果相互参照/一起训练 （空间结构存在相关性）**
   1. **叠加一个分割后的3值图作为第四通道训练**
   2. 或叠加单通道3值图滤镜
8. 输出预测结果，找到异常值最大的类型， 处理

**Task3**

1. Data Augmentation √
2. Re-write Dice Metric √
3. combine Dice, cup-disc ratio: Mean Absolute Error (MAE), CE Loss as Loss function
4. Baseline UNet
   1. ~~像素值为0的黑边影响训练效果 → 自适应赋值（阈值范围）~~ **~~?~~**
5. attention – GR-ConvNet √
   1. 精度>0.9 保存
   2. 512， GR-ConvNet channel=64
   3. CELoss and Dice Loss alpha =0.1/0.2
   4. Incremental Training without color jitter
   5. **~~Error:~~** ~~When do Incremental Training (GR-ConvNet, 0.9538, SGD), there were wrong results like ?~~

~~[EVAL] iter=100/40000 avg\_loss=-0.2188 dice=1.2188~~

~~[EVAL] iter=200/40000 avg\_loss=1.0841 dice=-0.0841~~

1. 视杯视盘一起分割，**强约束是视杯一定在视盘内**
   1. **利用该特性设计训练后的减法**
   2. baseline UNet貌似是把3个class分开训练的? 考虑在GR-ConvNet训练时添加该强约束
2. **baseline UNet貌似是把3个class分开训练的**？ —— 考虑必要性
3. utilize higher resolution to training:

最终结果256×256 → original resolution (512×512) → calculate scores 是否会有损失？

solution：提升输出结果 image\_size = 512/1024

报错（512）memory overflow?

1. **Post-Processing** - the prediction values are not in {0, 1, 2}
   1. **与GR-ConvNet 热图的获取有关？**
2. **multi-scale** Learning: combine local region
3. Paddleseg

!pip install paddleseg

from paddleseg.models import BiSeNetV2

1. **model ensemble**
2. mask R-CNN
3. resnet as feature detector + unet
4. efficinetnet\_unet
5. ~~由粗到细的分割~~

**Overall**

1. 全集训练： After在Train-Val Set 模式下选择模型，确定收敛iters, 使用全训练集训练
2. 全集训练时，可以减小保存模型的时间间隔evl\_interval，并保存保留**同一网络**训练过程中储存的验证集上预测精度的**top n**模型，求预测平均值
3. **save log**
4. model ensemble
5. **TTA (Test-Time Augmentation)** 对测试结果做增强，预测。集成结果

**预测结果集成方式**：平均，多数投票，取最值

1. **量化、裁剪和蒸馏 model compression** – paddleSim:
   1. Pruning
   2. Fixed Point Quantization
   3. Knowledge Distillation
   4. Neural Architecture Search (NAS)
2. champions’ experience
   1. **data cleaner**
   2. learning rate schedule
   3. from low resolution to higher resolution

**Learning rate schedule in Adam**

*scheduler = paddle.optimizer.lr.NoamDecay(d\_model=0.01, warmup\_steps=100, verbose=True)*

*adam = paddle.optimizer.Adam(learning\_rate=scheduler,…)*

cell 43 修改随机因子 random\_state

cell 67 保存模型的时机 if iter % evl\_interval == 0

**Official Guidance**

**预处理：data augmentation, 3D-OCT加阈值分割的mask作为inputs，裁剪/阈值分割2D彩照黑边**

**Ensemble:**

保留**同一网络**训练过程中储存的验证集上预测精度的**top n**模型

**不同网络**训练获得的模型，测试，综合结果

**TTA (Test-Time Augmentation)** 对测试结果做增强，预测。集成结果

**预测结果集成方式**：平均，多数投票，（取最大

**量化、裁剪和蒸馏**

**多任务协同训练：**

**黄斑中央凹定位**

可转化为分割任务训练

~~可与对应结构的分割任务一起训练~~

可转化为**目标定位**任务训练（采用**高斯**等**得到一个范围**）- 参考GR-ConvNet得到抓取中心点

**可与视盘分割或定位任务一起训练（空间结构存在相关性）**

**视杯视盘分割：**

视杯视盘一起分割，**强约束是视杯一定在视盘内**

~~分割任务与对应结构中心定位任务一起训练~~

~~视杯视盘分割与黄斑定位一起训练~~

**由粗到细的预测过程：**

黄斑中央凹定位：先粗略定位/分割黄斑区域，然后精确定位中央凹

视盘/视杯分割：先粗略确定视盘区域，可裁剪出该区域，然后精细分割视盘/视杯

**将task2, task3的结果/网络/训练特征图用于task1**

**可自行标注本比赛提供的数据，如视神经纤维层厚度等**

**不得使用任何其他渠道的眼底标注数据（包括且不限于已公开的眼底数据集、自行标注的数据集等）**

**自然图像的训练结果提取图像特征，e.g. ImageNet √**

**自监督提取图像特征 ？**

**后处理**