1. **选择这个课题的原因是什么？**

这个课题是我的指导老师推荐我做的，三维模型的应用越来越广泛，也建议我和深度学习方面的知识结合一下，做一些改进。

1. **该课题研究的意义和目的？**

现在每天都要创建大量的三维模型，完全从零开发耗时耗力，能从现有的库里找到合适的模型是比较重要的，项目提供了一个检索的平台，可以高效地检索。

1. **全文基本结构、框架是怎么设计的？**

文章的结构也是我毕设实现的顺序，是按照项目背景——核心算法实现——项目功能实现——测试与总结的顺序写的。

1. **你在写这篇论文的时候遇到的一些困难和如何解决的？**

主要是学习资料不足，很多用到的工具都是没有中文的使用教程的，只能去读他官方给出的API手册。

1. **论文的创新点在哪里？**
2. 算法改进提升了检索精度
3. 可拖拽缩放，用户体验好
4. 安全性好，为商用或工业化做准备。

**六、主要工作量在哪儿？自己做了什么？**

自己写的摘要读一遍

由于深度学习很多都是共用的，一部分是之前做的项目，一部分是已有的代码，核心算法部分70-80%，系统功能100%。

**PPT里的专业知识**

1. **讲一下维数灾难**

维数灾难是一种过拟合的现象，是由三维模型的数据结构特性造成的，比如用一张二维图片作为输入，256\*256的图片才65000个维度，相当于解决一个65000维空间的问题。但是三维模型为了表示空间中点和点的关系，单个三维模型的数据量很大，直接输入就导致了特征维度非常多。在样本量固定的情况下，样本数和特征维度不呈一个正比的指数递增关系，就会导致分类器性能很差，表现为过拟合。

最后用的是227\*227\*3，150000多个维度。

1. **数据集怎么做？预处理？**

数据集我是自己写了脚本调用了blender软件里的库函数实现的，这个软件是专门对三维模型做处理的，由于它完全开源所以我可以通过api来直接调用里面的一些函数，就相当于调用opengl库函数。

1. **共享权重？**

**用的是tf.variable\_scope()函数，因为网络结构完全一样，所以参数不需要重复设置。**

1. **什么是卷积层？什么是全连接层？什么是池化层？**

卷积层就是，我的理解是一种特殊的数学运算或者函数，使用多个滤波器在源数据上进行卷积操作，提取特征。

全连接层是一种特殊的卷积层，他用的是1\*1的滤波器，步长为1，不共享参数，主要起一个分类的作用。

池化层分两种，我打个比方，2\*2的一块数据，最大池化就是取这四个数据中的最大的一块，平均池化就是取这四个数的平均。

1. **View-polling层？**

这一层相当于一种特殊的池化层，这层的输入输出尺寸是12\*9216到9216，就是把12层feature map堆叠起来取12层中每一层的最大值。

因为12条分支总要汇合，原论文尝试了取平均值操作发现效果很差，取最大值效果就非常好了。

1. **分类器？损失函数？**

Softmax函数是分类函数，把fc层传过来的数据求偏导然后归一化到0-1的区间来判断类别。和SVM的区别在于：SVM更注重正确的类别，只要过了某个阈值，就判定为某类别，softmax会计算样本在所有类别上的概率。

损失函数用的是sparse\_softmax\_cross\_entropy\_with\_logits函数，是一种简化的交叉熵函数，可以直接对标签和feature map做运算。第一步先做softmax处理，第二步是计算交叉熵，主要针对多类别的分类问题做了优化，也是tensorflow的内置函数。

1. **Batch-size？**

16\*12,12是12张图片。

1. **Relu作用？**

首先这种激活函数是非线性的，相较于线性函数，可以拟合几乎所有问题

其次加快收敛速度，relu计算起来更快

负区间为0，直接抑制一部分神经元，增加了神经元的稀疏性，缓解过拟合

非负区间的梯度是常数1，无梯度消失现象。Sigmoid函数是一个s型的曲线，在后面饱和区的导数接近0，收敛的会很慢。

1. **LRN层和dropout区别？**

Lrn层在后来的表现效果一直不太好，被舍弃了。Dropout舍弃了50%的神经元，减少了参数量，起到了防止过拟合的效果。

1. **mAP？**

mAP值是平均检索精度，单个样本检索精度是，对所有检索到的文档编号除以他在结果列表中的位置求和取平均，mAP是所有的AP取平均。mAP的结果对位置信息比较敏感，非常适合检索类别的标准。

1. **没加载模型材质？**

为了去除颜色和纹理干扰

1. **H5文件？**

read csv：28s 12311

read h5:4.5s

1. **为什么选B/S？**

美观、方便部署到服务器上、系统更新无需下载

1. **MVC讲解？**

Model：核心检索算法与其他的处理代码。

View：Web界面，渲染数据。

Control：Flask 框架路由，负责响应用户请求、准备数据、前后端交互。

1. **Flask框架好处？**

相比于Django更轻量，没有太多的用不到的功能，易于维护。区别也是Django更大，功能更全面，适合大型网站开发。

1. **为什么划分权限？**

为了系统安全性，不能随便上传模型，随便使用服务器和显卡资源，也更符合工业化和商业的使用场景。

1. **用的什么测试方法？**

黑盒测试里的等价类划分法

1. **安全性体现在？**

管理员权限、

密码加密存储、

SQL防注入、

上传时间间隔、

session防止url跳转。

1. **数据库加密算法好处？原理？**

密码加密算法常使用单向的 HASH 算法比如md5算法。哈希算法虽不可逆，但是每次加密的结果是一样的，只需要根据加密表进行反向解析就可以推测出密码明文， Bcrypt 算法引入“加盐”机制，密文是由密码 password和一段完全随机的信息盐 salt 进行多次 hash 操作得来的，验证的时候用用户提供的密码和这段随机信息再次哈希，看是否相同。

一个使用 bcrypt 加密后的密码，用暴力穷举的方法破解甚至需要几十年时间。每次得到的密文也不同，防止了彩虹表的添加和收录。

1. **数据库变长长度？**

20、60

**视频里的功能如何实现**

1. **前后端交互实现？**

用了ajax发送post和get请求实现的，登录和上传文件是post，访问资源是get。

1. **Post和get区别？**

Get会把参数放到url里传输，Post会传输一个request body，更安全一点。

Post没有传输格式限制，get只能传ascll码。

Post发两次tcp请求，get一次。

1. **分页？**

缩略图是9个一组的，所以把模型的地址列表九个九个分组，翻页时找到相应的地址组然后渲染模型。

1. **如何渲染3D模型？**

使用了Three.js这个三维模型web开发引擎，这个引擎是基于webgl封装的，提供了三维模型在网页端的渲染、缩放、选择、拖拽等功能的api。

1. **如何下载？**

访问服务器url

1. **模糊检索的实现？**

使用了boostrap-select控件，支持项目的模糊搜索。

1. **上传到数据库限时的实现？**

在后台设置了上传间隔的计时器，请求服务先判断是否间隔十秒。

1. **后台和数据库怎么连接？**

使用了python的pymysql包，先建立连接，获得游标，然后给游标绑定sql语句，用excute执行，excute还能防止sql注入。

1. **为什么选MySql？**

比较通用、还能防注入，excute对特殊字符进行了加\转义处理。

1. **格式转换的实现？**

调用Blender的库函数，blender应该是基于OpenGL做了封装。

1. **Session？防止url跳转？**

Session是会话，密码验证成功之后会把登录用户的信息存到浏览器里，如果直接访问URL没有信息会拒绝访问。

1. **删除和入库怎么实现？**

删除就是删除目录下的模型文件和特征库里的特征，入库同理。

1. **模型在线训练为什么只能多GPU？如何保证效果？怎么检索新模型？**

在线训练需要显卡资源，服务启动也要显卡资源。90%以上自动停止，管理员可以后台看代码，管理员替换数据源。

1. **是否支持多线程？如何保证多线程资源互斥？**

Flask框架天然支持多线程，每个用户之间会加资源锁处理。

**十五、没有注册？**

没有，这个功能比较简单，而且也不是我系统的主要任务，所以就没有做，如果想添加用户可以在后台直接修改数据库，添加一条记录就可以了。

结果问了：

1、 解释一下学习率的含义，学习率初始就是0.0001吗？是直接从0.001设置为0.0001的吗？没有自动设置学习率的改变吗？

答：学习率直接关系到梯度下降的速率，学习率设置得较高会导致收敛速度加快，设置得低会放缓梯度下降的速率，但是会更容易找到梯度最优点，loss值也会越小。我是先设置的0.001的学习率，同时保持了0.1的指数型衰减。在loss下降不明显的时候换为0.0001的学习率，同时也保持了0.1的指数型衰减。

2、 dropout是一开始就用了吗，效果怎么样，对比了吗？

答：是一开始就用了dropout，原论文使用的是Lrn层来做防止过拟合的处理，我直接使用dropout替代了Lrn层，最终分类准确率达到了91.86%，比原论文提升了2.3%。平均检索精度达到了87.58%，比原论文提升了7.3%。

3、三维模型的存储的模式是怎样的？是点云吗？存储需要多少空间呢，有多少个维度？会产生维度爆炸问题吗？用了什么技术来解决呢？

答：三维模型是以obj文件或者off文件的格式存放的，不是点云。大概一个10M的三维模型会产生130000多维度的特征空间。三维模型的样本数量是很少的，样本数量和特征维度不呈一个正比的指数型增长关系，就会引发“维数灾难”，这也是一种过拟合的表现。