附件1：

**软件学院创新项目实训**

**申请表**

**（2024版）**

项目编号 （申请人不必填写）

项目名称 Sapphire：基于SAM的智能辅助数据标注平台

团队负责人 侯玉祺

手机号码 15889766016

项目时间 2024年3月20日 至 2024年6月23日

山东大学软件学院

**填报说明**

1. **项目目标**请简短说明项目的研发定位。
2. **指导教师**请暂时不需要填写。
3. **技术点**请列出项目中使用的关键技术点，最多不超过5个名词。
4. **参加人员**中的第一行必须为项目负责人。团队总人数4-5人，少于3人、超过5人（小于等于3人，或者大于等于6人）需要提前发邮件申请。所有成员必须为2021级软件学院的大三在校本科生，其他参与人员请在其他人员中列出。
5. **项目资料地址**必须可公开访问，项目组每个成员都需要维护各自的博客，以便督导老师随时可以抽查项目实施过程。必须选用新浪、CSDN等知名服务提供商，不支持自己搭建的服务器。
6. **项目介绍**从项目背景、技术创新、工作内容、技术路线、实施方案等方面，详细说明项目的具体工作。请详细说明，可以添加附件，或者音视频等多媒体材料的网址。
7. **实施计划**按照时间节点，分阶段完成预定目标。
8. 申请承诺需要团队所有成员签字。

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **项目名称** | Sapphire：基于SAM的智能辅助数据标注平台 | | | | | |
| **实施时间** | 2024 年3月20日 至 2024年6月23 日 | | | | | |
| **项目目标** | 基于Segment Anything Model(SAM)的先进图像分割技术，结合众包任务分发概念，搭建AI辅助数据标注平台，填补细分领域的市场空白。 | | | | | |
| **技术要点** | 数据标注，大规模多模态预训练分割网络，SAM通用视觉大模型，众包，端云结合 | | | | | |
| **指导教师** | （先不填） 手机号码： | | | | | |
| **团队名称** | 重生之我在山大标数据 | | | | | |
| **参加人员** | **姓名** | **学号** | **专业** | **手机号码** | **电子邮箱** | **项目分工** |
| 侯玉祺 | 202100300341 | 软件工程 | 15889766016 | houyuqi2003@foxmail.com | 组长，负责人 |
| 吴羲勇 | 202100300261 | 软件工程 | 19963645235 | wxyww2049@gmail.com | 代码、文档编写 |
| 朱哲甬 | 202100300082 | 数字媒体技术 | 18967892613 | zhuzy@mail.sdu.edu.cn | 代码、文档编写 |
| 郭苏睿 | 202100300224 | 软件工程 | 13012615613 | 13012615613@163.com | 代码、文档编写 |
| 伍钰棋 | 202102300398 | 数字媒体技术 | 19168852517 | irra.c923@gmail.com | 代码、文档编写 |
| 其他人员 | | | | | |
| **项目资料地址** | | | | | | |
| **项目博客地址：**https://blog.csdn.net/m0\_59025440/category\_12621020.html  **Gitee地址：**https://gitee.com/wxyww/sapphire  **成员个人博客地址：**  侯玉祺：https://blog.csdn.net/ellyi/category\_12620990.html  郭苏睿：https://blog.csdn.net/qq\_40068267/category\_12620988.html  吴羲勇：https://blog.csdn.net/wxyww2049/category\_12621009.html  伍钰棋：https://blog.csdn.net/xi\_irra/category\_12621018.html  朱哲甬：https://blog.csdn.net/u012185664/category\_12621022.html | | | | | | |
| **项目介绍（可跨页，可附加多媒体网址）** | | | | | | |
| 1 项目背景1.1 研究背景 在人工智能和机器学习领域，数据标注的重要性不言而喻。这是训练准确模型的基础，直接影响到模型的性能和应用效果。然而，传统的手动数据标注方法效率低下，成本高昂，且难以保证一致性和准确性。随着数据量的爆炸性增长，这些问题愈发突出。  **技术角度：**  随着深度学习和其他人工智能技术的发展，数据标注不再局限于人工操作。现代技术使得数据标注过程可以通过算法自动完成，大幅提高了效率和准确性。例如，AI预标注技术可以在语音转写项目中自动完成语音识别和文字转写的预处理操作，之后只需人工进行少量校对。此外，质量控制也逐渐由人工审核转向AI质检，通过算法评估标注结果的准确性。  例如，**SAM（Segment Anything Model）模型**作为Meta AI推出的一种创新的图像分割模型，能够在无需训练的情况下对任何目标进行分割，为自动化数据标注提供了新的可能性。这也就大大降低了图像领域的标注门槛。  **市场角度：**  数据标注市场规模正迅速扩大，并呈增长趋势。据澎湃新闻的报告，中国的AIGC数据标注市场规模已达到百亿量级，且预计未来几年将持续增长，到2025年市场规模将突破100亿元。    同时，数据标注行业的竞争格局也正在发生变化，大模型公司和AI企业正在涌入这一领域。自建数据处理管线，提供一站式、定制化的数据解决方案，是数据行业的大势所趋，具有巨大的市场需求和商业机会。  综上，随着技术的进步和市场的扩大，数据标注正逐渐从一个辅助性角色，转变为推动人工智能发展的关键力量。 1.2 现存产品 目前，市场上的数据标注产品主要分为两类：**传统的人工标注服务和基于AI技术的自动化标注服务**。  传统的人工标注服务通常依赖于大量的标注人员手动处理数据。优点在于能够处理高度定制化的标注任务，然而，它们的缺点也很明显，包括成本高、效率低、扩展性差等。  基于AI的自动化标注服务则利用机器学习模型来自动识别和标注数据，优势在于能快速处理大量数据，但在处理复杂任务时的准确性可能有所欠缺。  我们的项目——**Sapphire：基于SAM的智能辅助数据标注平台，基于Segment Anything Model(SAM)的先进图像分割技术，结合众包概念和冗余任务分发模式**，为用户提供了高效、准确、低成本、易扩展的数据标注解决方案，还提供用户友好的界面和管理系统，填补了该细分领域的市场空白。 1.3 解决方案 近年来，自动化数据标注技术和众包平台的发展为数据标注领域带来了革命性的变化。  **自动化标注技术**通过算法和模型的应用，能够自动提取和标注数据，显著提高了标注的速度和效率，同时也降低了人工成本。其中，以SAM通用视觉大模型最为出色。  [**附件2：Segment Anything(SAM)模型调研报告**](#附件2)**中详细描述了这部分。**  **众包平台**则利用互联网的力量，集合了全球范围内的人力资源，不仅提升了数据标注的质量，还能够快速适应多变的项目需求。  在传统数据标注的基础上，本平台的设计理念是将人工智能与人类智慧相结合，以达到数据标注的最优效果。本项目有以下特点：   * 设定管理员与个人用户双角色。管理员负责任务发布和管理，标注者参与数据标注，个人用户可以是数据提供者或标注结果的使用者。 * 引入众包概念，允许广泛的用户参与标注过程，提升数据多样性和标注质量。 * 实现了数据标注任务的发布、分发、收集、分析、统计、管理等功能。 * 通过AIGC辅助标注，提高标注效率及正确率。 * 通过不同用户标注同一图片、对比结果的方式，确保标注结果的准确性和一致性。 * 兼有激励机制，增加用户黏性。  2 目标用户 本项目的目标用户群体非常广泛，涵盖了需要高质量数据标注服务的各个领域。包括但不限于以下几类：   1. **研究机构和学术团队：**需要大量标注数据来进行机器学习和数据分析研究的团队。 2. **AI公司和初创企业：**寻求提高数据处理效率，降低成本，并快速迭代其AI模型的企业。 3. **医疗健康行业：**需要精确标注医学图像数据以辅助诊断和治疗的医疗机构。 4. **自动驾驶汽车公司：**依赖于大量精确标注的传感器数据来训练其自动驾驶算法的公司。 5. **公共安全和监控：**需要自动化分析和标注监控视频数据的政府或安全机构。 6. **电子商务平台：**需要标注商品图片以提升搜索和推荐系统准确性的在线零售商。   通过为这些用户群体提供一个强大且易于使用的自动数据标注平台，可以极大地推动相关领域的研究和应用发展，同时也为平台的持续改进和创新提供动力。 3 项目内容 **Sapphire：基于SAM的智能辅助数据标注平台系统**包括以下五大模块： 3.1 用户系统 用户系统是本数据标注平台的基石，负责处理用户的注册、登录、角色分配、权限管理及个人信息维护等功能。用户系统应兼具可用性和安全性。 3.1.1系统角色定义  * **系统管理员：**有用户管理权限、可以发布官方任务、查看全局统计信息 * **众包发布者：**可以发布任务、发布时需指定对于任务投入的积分 * **工人：**认领任务者，上线后可以即时认领任务开始标注，每张图片有其价格（与发布者投入的积分有关）   用户角色既可以认领任务也可以发布，也就是既可以做发布者也可以做工人。 3.1.2 用户端功能  * **注册与登录：**新用户可以通过邮箱等注册，获取个人用户身份。注册后，用户可以登录平台，开始参与数据标注或管理标注任务。 * **个人资料管理：**用户可以编辑个人信息，如联系方式、密码等。 * **标注任务接收和提交：**接收、参与和提交标注任务。用户可以根据自己的兴趣和技能匹配任务。 * **标注工具与界面：**使用平台提供的多种标注工具和友好的界面进行数据标注。工具可以根据不同任务类型，如目标检测、分割等，进行相应的定制。 * **数据集浏览与下载：**用户可以花费积分等，获取平台提供的数据集。 * **激励机制：**积分与排名   1. 完成任务可获得积分，获取数据集需消耗积分   2. 用户可以查看自己的积分，参与平台排名竞争。   3. 标注质量保证：用户需要按照规定的标注准则进行标注，并确保标注结果的准确性和一致性。标注质量呈现在积分中。  3.1.3 管理员端功能  * **用户管理：**管理用户账户，包括注册审核、权限分配、用户数据统计等。 * **任务发布与管理：**管理任务的生命周期，包括任务创建、任务分发、任务进度监控等。 * **数据集管理：**创建、维护、删除数据集等。 * **数据质量保证：**制定标注准则和质量评估方法，确保标注结果的准确性和一致性。 * **分析统计：**查看和生成各类分析统计报告，包括用户维度、数据集维度和全局维度的统计数据，进行数据分析和决策。 * **系统监控：**监控系统状态，确保平台稳定运行。 * **激励机制管理：**管理员可以设置和调整激励机制，如积分规则和排名奖励。  3.2 分发系统 **系统定义：**  分发系统面向所有工人。由系统提供算法对所有待分配的任务进行分配。分发系统是众包系统的核心之一，需要设计算法准确、迅速的发布众包到具体工人，并及时更新任务标注状态。  **系统目标：**   * 高效   分发系统必须是高效的，当用户上线选择众包并开始标注后，系统必须能够迅速的做出响应，派发给用户需要标注并且未被当前用户标注过的数据。   * 正确率保证   分发系统还需保证标注的一致性。为保证标注的正确性，每张图片需要至少2名工人标注。系统需要设置恰当的算法比对两人标注结果的一致性，当标注差距超过预定阈值时，需要继续将此图片派发给第三个工人，若第三人与前两者标注不一致，则需继续派发、采取投票策略，超过5人仍无法确定时，图片标记为异常图片。   * 一致性   系统必须保证在多用户同时标注同一数据集的情况下仍能做出正确派发策略，确保每张图片被预定的人次标注。例如图片A需要两人标注，那么当10人同时上线时，不能将A同时派发给此10人。   * 预分发   为保证分发的效率与一致性，当用户认领时，会立刻在后台预锁定一批图片，锁定状态当作已标注处理。在用户标注时候，需要设置缓冲区，提前将将要标注的图片返回给工人，提升工人标注体验。   * 后处理   任务完成后，系统会触发后处理工作。对工人派发积分奖励，计算工人标注准确率并交给激励系统处理。最终将处理好的数据集放入后台，由任务发布者下载使用。  **分发策略：**    分发系统的数据流图如上。用户通过请求系统向数据库请求任务，请求系统会根据用户选择的数据集、用户已标注过的数据从数据库中选择数据进行分发，选择时会采取先锁定再分发的策略，由请求系统预锁定一批任务，然后交由用户标注，用户标注完成后在DB中解除锁定并改为已标注状态。流程图如下：    **质量管控：**  质量管控系统负责标注结果的正确性监测与保证。如果某任务非第一次标注，则与之前标注的结果进行比对，当差异超过阈值时，则需要下次派发，否则标注结束，进行后续工作，后续包括对标注工人评价，存入数据库，计算最终可训练数据等。    **性能优化：**  为了提供更优的性能，采取后台加载的策略，当用户进行数据标注的同时，从后台拉取预锁定图片，可以减少用户的等待时间。同时再前端设置缓冲区，缓冲区未慢时，从后端拉取任务，如果已满则暂停拉取。 辅助系统  * **领航者模式（Copilot Mode）**   用户提供初步的点或框图标注，SAM模型基于这些信息进行识别和标注。这种模式适用于需要精确位置信息的标注任务，如物体检测或图像分割。   * **半自动模式（Semi-Automatic Mode）**   半自动模式中，SAM模型根据给定的文本进行初步标注，然后由人工进行核查和修正。这种模式结合了AI的高效率和人类的判断力，适用于需要文本理解和验证的标注任务。   * **全自动模式（Fully Automatic Mode）**   全自动模式利用GPT vision作为中介，以解决SAM（Segment Anything Model）在文本识别方面的不足。在这种模式下，AI系统处理复杂的语义信息，并由GPT完成性能评估。这种模式适用于需要处理复杂语义和大量数据的场景，可以实现高度自动化的数据标注。 3.4 分析统计 **用户维度：**   * + 用户角色与权限：分析不同用户在平台上的角色和权限，例如标注员、管理员等，以及他们在平台上的操作行为和需求。   + 用户活跃度：分析用户在平台上的活跃度，包括登录频率、标注次数等，了解用户参与度和贡献度。   + 用户反馈：收集用户反馈数据，了解用户对平台功能、界面和服务的满意度，以及可能的改进建议。   **数据集维度：**   * 数据标注质量：分析数据标注的质量指标，如准确性、一致性和完整性，评估标注结果的可靠性。 * 数据集分布与特征：分析数据集中不同类别或类型数据的分布情况和特征，为标注任务的设计和优化提供依据。   **全局维度：**   * 平台整体效率：分析整体平台的运行效率，包括标注任务的完成速度、审核流程的效率等，以提高整体工作效率 * 数据利用与应用：探讨标注数据的应用领域和潜在价值，分析标注数据对业务决策和模型训练的影响。 * 模型评估和优化：评估建立的模型在全局数据上的效果，并根据评估结果对模型进行优化和调整。  3.5 激励机制 **排名系统：**   * + 标注员排名：根据标注员的标注质量、效率和参与度等指标，设立排名系统，对表现优秀的标注员进行排名展示。   + 数据集质量排名：根据数据集的质量评估指标，对数据集的质量进行排名，激励用户创建高质量的数据集。   **积分系统：**   * + 任务积分奖励：为达到标注任务标准线的标注元提供积分奖励，标准线设置为每位标注员的平均标注积分，未达标的标注员会扣除相对应的积分。积分可以用于兑换奖励或提升在平台上的身份和权限。   + 质量奖励积分：根据标注质量评估结果，奖励标注质量高的标注员额外积分，鼓励提高标注质量。  3.6 数据集系统 **创建数据集：**   * 数据集需求分析：管理员提供数据需求和标注任务类型，根据商户需求发布不同类型的数据集。   **发布数据：**   * 数据集发布：   + 用户（小量数据集）消耗积分在平台内上传数据集，标注员直接接单，完成标注。   + 商户（大量数据集）向平台提供数据集，管理员接收后设置积分，发布成官方任务，标注员可以接单完成标注。 * 数据集权限管理：设定数据集的访问权限，包括公开数据集、私有数据集和共享数据集，以保护数据安全和隐私。   **创意工坊：**  是一个专门的区域，让用户分享关于数据标注任务的创意、经验和解决方案。   * + 个性化账号：用户可以通过积分兑换系统，定制个性化账号信息、个人背景和边框装饰等，从而增强其在平台上的个性化展示和参与动力。   + 界面设计：设计直观友好的界面，让用户可以轻松浏览、搜索和参与创意工坊中的内容和讨论。  4 前端技术路线4.1 核心需求概述 本系统采取重前端轻后端的实现框架，由于模型大小较小，所以可以使用浏览器提供的onnx运行时进行推理。在模型推理交给前端的前提下，前端的核心需求如下   * 模型推理 * 用户多种输入捕获 * 推理结果展现与格式化  4.2 关键技术 最主要的关键技术为onnx运行时的调试和使用，需要验证onnxruntime-web的全套输入输出链。  以模型的推理为核心，在推理前需要获取图片的npy数据，由于npy的计算对算力要求较高，所以交由后端实现，之间传回前端。  模型的输入还有用户侧的输入，需要验证用户点击、框选两种输入模式下对于模型的调用情况。  在推理后需要对于模型的输出进行处理，一方面，需要形成mask在用户侧进行展示，另一方面，需要将数据化的结果传递给后端。 4.3 技术验证4.3.1用户侧输入验证 首先对于用户输入侧的数据捕获进行验证。由于框选可以视为捕获down和up时的不同坐标，所以与点击使用的技术相同，这里只对点击进行验证。  编写工具类监听鼠标坐标变化并在控制台输出：     4.3.2 onnxruntime-web 对模型的运行进行验证，运行需要指定模型、npy。为了便于技术验证，暂时将所有文件都放在本地。  编写模型的输入处理：  clicks为鼠标点击坐标，tensor为加载进来的npy文件，然后传入模型尺寸。下面的方法会输出一个模型输入用的参数。  const modelData = ({ clicks, tensor, modelScale }: modeDataProps) => {  const imageEmbedding = tensor;  let pointCoords;  let pointLabels;  let pointCoordsTensor;  let pointLabelsTensor;  if (clicks) {  let n = clicks.length;  pointCoords = new Float32Array(2 \* (n + 1));  pointLabels = new Float32Array(n + 1);  for (let i = 0; i < n; i++) {  pointCoords[2 \* i] = clicks[i].x \* modelScale.samScale;  pointCoords[2 \* i + 1] = clicks[i].y \* modelScale.samScale;  pointLabels[i] = clicks[i].clickType;  }  pointCoords[2 \* n] = 0.0;  pointCoords[2 \* n + 1] = 0.0;  pointLabels[n] = -1.0;  pointCoordsTensor = new Tensor("float32", pointCoords, [1, n + 1, 2]);  pointLabelsTensor = new Tensor("float32", pointLabels, [1, n + 1]);  }  const imageSizeTensor = new Tensor("float32", [  modelScale.height,  modelScale.width,  ]);  if (pointCoordsTensor === undefined || pointLabelsTensor === undefined)  return;  const maskInput = new Tensor(  "float32",  new Float32Array(256 \* 256),  [1, 1, 256, 256]  );  const hasMaskInput = new Tensor("float32", [0]);  return {  image\_embeddings: imageEmbedding,  point\_coords: pointCoordsTensor,  point\_labels: pointLabelsTensor,  orig\_im\_size: imageSizeTensor,  mask\_input: maskInput,  has\_mask\_input: hasMaskInput,  };  };  然后，根据捕获到的数据进行推理，模型输出是一个灰度图，需要将数组数据转化成图片数据。  const feeds = modelData({  clicks,  tensor,  modelScale,  });  if (feeds === undefined) return;  const results = await model.run(feeds);  const output = results[model.outputNames[0]];  setMaskImg(onnxMaskToImage(output.data, output.dims[2], output.dims[3])); 4.3.3 mask图生成 模型的输出一个灰度图，需要转化成mask的图片遮罩在原图片上。下面的方法将灰度图中所有为大于0的位置全部赋予预定的蓝色。这样就形成了一层mask图的数组。然后使用imageDataToImage将数组数据转化成一张图片即可。最后将mask图遮罩在原图上。  function arrayToImageData(input: any, width: number, height: number) {  const [r, g, b, a] = [0, 114, 189, 255]; // the masks's blue color  const arr = new Uint8ClampedArray(4 \* width \* height).fill(0);  for (let i = 0; i < input.length; i++) {  if (input[i] > 0.0) {  arr[4 \* i + 0] = r;  arr[4 \* i + 1] = g;  arr[4 \* i + 2] = b;  arr[4 \* i + 3] = a;  }  }  return new ImageData(arr, height, width);  }  function imageDataToImage(imageData: ImageData) {  const canvas = imageDataToCanvas(imageData);  const image = new Image();  image.src = canvas.toDataURL();  return image;  } 4.3.4 验证效果 最终验证效果如下。通过点击可以传输给模型位置，模型输出的数据形成mask图后可以正确的遮盖在原图上。   5 后端技术路线5.1 核心需要概述 本系统中，后端的核心需求主要包括：   * 提供用户注册、登录、登出等用户管理功能 * 提供模型嵌入、模型管理的功能 * 任务发布、任务分发功能 * 统计数据收集、分析  5.2 关键层 后端使用***Spring Cloud Alibaba***框架开发，预期使用的组件包括但不限于：   * Spring Boot Actuator * Spring Boot Validation * Nacos Discovery * Nacos Config * Spring Cloud Loadbalancer * Sentinel * OpenFeign  5.2.1 网关层 业务网关层使用***Spring Cloud Gateway***，提供统一的API入口，对请求进行路由、限流、鉴权等操作。  Nginx 网关负责服务器端的转发与负载均衡。 5.2.2 业务层 业务应用使用***Docker***进行容器化管理，搭建CI/CD执行自动编译，Shell 脚本打包***Docker Image***和更新。根据业务规模酌情考虑多台机器上的业务分布。 5.2.3 持久化层 数据库使用***PostgreSQL***，提供数据存储服务。多节点部署，保证数据的高可用性，同时集群化部署，热备份，并且实现读写分离，保证数据一致性。  采用***Redis***作为缓存数据库，提供缓存服务，减轻数据库的压力。同时，在分布式系统中，使用***Redis***作为分布式锁，保证数据的一致性。  所有持久化数据至少都保证主从备份。 5.3 技术验证5.3.1 网络验证 从两三个节点开始，架设至少主从的网络拓扑结构。对于每个节点，都使用ufw进行防火墙配置，只开放必要的端口。  sudo ufw default deny incoming  sudo ufw default allow outgoing  这样以后，自主放行一些incoming流量。  最后***sudo ufw enable***，开启防火墙。 5.3.2 中间件部署验证 在不同服务器上使用 Docker 部署各个中间件，测试各个中间件的可用性。    主从配置例子如下：  docker run -d \  -p 5773:5432 \  -e POSTGRES\_PASSWORD=EM45rjPbymGHLenq \  -v /data/postgres/master/:/var/lib/postgresql/data \  --name postgres-master \  --restart unless-stopped \  -h postgres-master \  -e LANG="C.UTF-8" \  -e 'TZ=Asia/Shanghai' \  -m 512m \  --memory-swap -1 \  postgres:15.3-alpine3.18  WAL（Write-Ahead Log）是 PostgreSQL 中的一种事务日志，用于记录对数据库的更改操作。它是一种持久化的日志，记录了对数据库进行的每个事务的详细更改。WAL 允许数据库在发生故障时进行恢复，并支持主从复制和逻辑复制等高级功能。通过持久化的 WAL 日志，PostgreSQL 可以保证数据的一致性和持久性。  ***wal\_level***可以是***minimal***，***replica***，***logical***，区别在于：   * ***minimal***只提供必要的信息以支持数据库的恢复，不支持任何形式的复制； * ***replica***：记录足够的信息以支持从服务器的物理复制。它包含主服务器上的所有更改，从服务器可以使用这些信息进行复制和恢复； * ***logical***级别的复制中，WAL 用于逻辑复制。逻辑复制允许选择性地复制特定的表和数据更改，而不是简单地复制整个数据库。   开启***archive\_mode***，PostgreSQL 将尝试执行归档命令来将 WAL 日志文件复制到归档目录。  验证主从连接，在 master 节点上，进入 PostgreSQL，执行 ***select client\_addr,sync\_state from pg\_stat\_replication;***，能够观察到形如下文的结果，即配置成功。  postgres=# select client\_addr,sync\_state from pg\_stat\_replication;  client\_addr | sync\_state  ---------------+------------  172.16.193.30 | sync  (1 row) 5.3.3 CI/CD 部署验证 项目使用 Jenkins 进行 CI/CD 部署验证，编写 Pipeline 脚本，实现自动化部署。  此外，同时利用 GitHub Actions 进行 CI/CD 部署验证。 5.3.4验证深度学习模型嵌入调用验证 在 Jupyter Notebook 中测试了小规模的深度学习模型，验证模型的准确性。   6 实施方案 **Sapphire：基于SAM的智能辅助数据标注平台**项目预计包含需求分析与产品设计、核心技术验证、网页端开发、服务端开发、测试与部署、文档编写等六个阶段。每个阶段的具体内容如下：   1. **需求分析与产品设计：**根据项目需求，分析项目的功能需求，设计项目的产品原型。 2. **核心技术验证：**验证项目的核心技术，包括图像识别、语音识别、自然语言处理等。 3. **网页端开发：** 搭建前端用户界面，包括页面布局、交互设计、逻辑编程等。保证系统的兼容性。 4. **服务端开发：**开发项目的服务端，包括后端服务、数据库服务等。 5. **测试与部署：**测试项目的各个模块，部署项目到服务器上。 6. **文档编写：**编写项目的各个阶段的文档，包括需求分析文档、设计文档、测试文档等。 | | | | | | |
| 实施计划 | **Sapphire：基于SAM的智能辅助数据标注平台**项目预计于本学期第六周开始，第十七周结束，共计十二周。  具体的实施计划如下：  第一周（3.25~3.31）：需求分析与产品设计。  第二周（4.1~4.14）：核心技术验证。  第三～第八周（4.15~5.26）：网页端与服务端开发。  第九～第十周（5.27~6.9）：测试与部署。  第十一～第十二周（6.10~6.23）：文档编写和项目总结。 | | | | | |
| 预期成果 | 实现一个完整的自动分发AI辅助数据标注平台。用户上线可直接认领任务开启数据标注任务。用户可以发布数据集，导出标注完成的任务。  系统应拥有：   1. 完善的质量管控系统，确保标注的数据经过多人次验证，差异值需符合设定阈值。 2. 合理的积分系统，发布数据集通过积分系统进行悬赏，完成标注工作依据正确率给予积分奖励。 3. 健壮的系统性能，在大量用户同时标注的状态下，确保服务的一致性和稳定性。 4. 精确的AI辅助系统，通过AI辅助用户可以快速的通过点击、粗框选完成数据标注工作。 | | | | | |
| 指导教师评语 | （先不填）  （签字） 年 月 日 | | | | | |
| 本人郑重承诺，此申请书内容真实有效。    （所有团队成员签字）  2024年3月27日 | | | | | | |

附件2：

# Segment Anything(SAM)模型调研报告

## 1 概述 Introduction

SAM是 Meta 提出的一种通用的、（准）多模态的 Zero-Shot 分割模型，极大地促进了计算机视觉基础模型的发展。

SAM是一个提示型模型，其在1100万张图像上训练了超过10亿个 Mask，实现了强大的零样本泛化。许多研究人员认为「这是 CV 的 GPT-3 时刻，因为 SAM 已经学会了物体是什么的一般概念，甚至是未知的物体、不熟悉的场景（如水下、细胞显微镜）和模糊的情况」，并展示了作为 CV 基本模型的巨大潜力。

SAM的输入与输出结构比较简单，其输入是一张图像和 Prompt，输出是图像的分割结果，与传统的分割网络相比并无太大区别。SAM 的核心在于输入的 Prompt，Prompt 是一种提示信息，用于指导模型进行分割，这也是 SAM 能够实现通用分割的关键。SAM 支持四种 Prompt 类型：***mask***、***points***、***box***和***text***，分别对应分割掩码、关键点、边界框和文本信息。不过，文本提示的效果并不理想，因此更多的使用 CLIP 配合 SAM 实现文本引导的分割任务。

在模型结构上，与现在大量 Decoder Only（以GPT为代表）的模型不同，SAM 仍然是一款建立在传统 Encoder-Decoder 结构、卷积神经网络的基础上的模型。SAM 包括三个主要结构：***Image Encoder、Prompt Encoder***和***Mask Decoder***，其中***Prompt Encoder***是 SAM 的核心，它将图像和提示信息进行编码，然后输入到***Mask Decoder***中，生成分割结果。

在实际应用中，SAM 通常可以视作 ***Image Embedding*** 和 ***Segment*** 两个任务阶段。在 ***Image Embedding*** 过程中，SAM 需要使用 ***ViT*** 对图像进行编码，这个过程是相对耗时的。而在 ***Segment*** 过程中，SAM 只需要对 ***Prompt*** 和 ***Image Embedding*** 进行编码，然后输入到 ***Mask Decoder*** 中，生成分割结果。因此，SAM 的推理速度相对较快。

## 2 研究背景Background

### 2.1 零样本分割 Zero-Shot Segmentation

视觉任务大致分为分类、检测和分割三类，其中分割是最为复杂的任务。传统的分割任务通常需要大量的标注数据，而零样本分割则是一种**无需标注数据**的分割任务。

零样本分割的关键在于模型的泛化能力，即**模型能够在未见过的数据上进行分割**。这意味着，对于 Zero-Shot Segmentation 任务，模型需要具备以下能力：

* **通用性**：模型能够对任意物体进行分割，而不仅仅是训练集中的物体。
* **泛化性**：模型能够在未见过的数据上进行分割，而不仅仅是训练集中的数据。

为了训练零样本分割模型，往往需要构建一个十分巨大、包含各种物体的数据集。然而，这种数据集的构建是十分困难的，因此 Zero-Shot Segmentation 一直是计算机视觉领域的一个热门研究方向。

### 2.2 预训练 Pre-training

随着神经网络结构的不断发展，预训练技术逐渐成为了计算机视觉领域的主流。预训练技术的核心在于**利用大规模数据集进行训练**，然后将训练好的模型迁移到其他任务上进行微调。

在传统图像处理任务中，ResNet、VGG 等网络都是典型的预训练模型，或者称为 ***Backbone***。这些预训练模型通常会在一个大的数据集上进行训练，具有良好的特征提取能力。通过使用 ***Backbone*** 对图像进行特征提取，并在此基础上添加新的模型结构进行训练，往往能获得更高的性能、缩短训练时间。

预训练解决的核心问题在于神经网络规模不断增大时，数据集和计算资源的需求也在不断增加。通过引入 Pre-training，大量的下游任务可以充分利用 ***Backbone*** 强大的特征提取能力再进行训练，有效降低了下游任务的模型规模和训练时间。

### 2.3 ONNX Runtime

神经网络的运行和部署是一个十分重要的问题。对于科研人员而言，***PyTorch***和***TensorFlow***等框架提供了便捷的训练和调试环境，但是这些框架往往不适合直接部署到生产环境中。

首先，这些框架往往对运行环境和硬件设备有一定的要求，而生产环境往往是多样化的，这就需要对模型进行一定的转换。其次，这些框架的运行效率往往不高，对于一些对速度要求较高的场景，这就需要对模型进行一定的优化。

在模型层面上，投入生产和部署时常常会进行量化压缩操作，以减少模型的体积和加速模型的推理速度。这是由于多数神经网络在训练阶段使用***fp32***或***fp16***的浮点数进行计算，而在推理阶段往往可以使用更低精度的整数（***int8***）进行计算，以减少计算量，并且对 CPU 和 GPU 的支持更好。

运行层面上主要有两种思路，分别是通用运行时和编译运行：

* + 通用运行时：通用运行时是一种支持多种硬件设备的运行时，如***ONNX Runtime。ONNX Runtime*** 是一个开源的深度学习推理引擎，支持 ONNX 格式的模型，并且支持多种硬件设备，如 CPU、GPU、FPGA 等。
  + 编译运行：编译运行是一种将模型编译成特定硬件设备的指令集，以提高模型的运行效率。这种方式往往需要对模型进行一定的优化，如量化、剪枝等。主要框架有***ncnn***、***MNN***等。

***ONNX*** 现在由微软维护，他的实现思路是通过记录模型的计算图，得到模型的计算过程，生成一个计算图的执行计划（类似***Java***编译后的字节码）；在设备上，通过 ***ONNX Runtime*** 加载模型，执行计算图的执行计划（类似***JVM***），从而实现模型的推理。