# 低资源语言的零样本 OCR 准确性：对僧伽罗语和泰米尔语的比较分析

Nevidu Jayatilleke and Nisansa de Silva
Department of Computer Science & Engineering
University of Moratuwa
Sri Lanka
{ nevidu.25, NisansaDdS } @cse.mrt.ac.lk

Abstract

由于在英语和其他高资源语言（HRL）上的大量研究，拉丁及其衍生字母的印刷文本的光学字符识别（OCR）问题现在可以被认为已经解决。然而，对于使用独特字母表的低资源语言（LRL），这一问题仍然亟待解决。本研究对六种不同 OCR 引擎在两种 LRL 语言：僧伽罗语和泰米尔语的零样本性能进行了比较分析。所选择的引擎包括商业和开源系统，旨在评估每类系统的优势。Cloud Vision API、Surya、Document AI 和 Tesseract 对僧伽罗语和泰米尔语进行评估，而 Subasa OCR 和 EasyOCR 仅对其中一种语言进行了检查，原因是受限于其功能。这些系统的性能使用五种测量方法进行了严格分析，以评估其在字符和词汇层面上的准确性。根据研究结果，Surya 在僧伽罗语中的所有指标上表现最佳，词错误率（WER）为 2.61 %。相反，Document AI 在泰米尔语中所有指标上表现突出，其字符错误率（CER）非常低，仅为 0.78 %。除了上述分析外，我们还引入了一种新的合成泰米尔 OCR 基准数据集 [1]。

## 1 引言

光学字符识别（OCR）是一种计算技术，用于识别数字图像中的文本，例如扫描的文档、广告和照片（Agarwal and Anastasopoulos, 2024; Jain et al., 2021; Weerasinghe et al., 2008）。OCR 通常被用作信息输入工具，从扫描的文档中提取有价值的数据，如表格、收据、发票和护照。历史上，OCR（连同文本转语音系统）是为帮助盲人或残疾人而开发的，通过机器为他们朗读书写的文本，这一发展可以追溯到 1914 年（Mittal and Garg, 2020）。

OCR 的过程通常包括多个步骤：1）首先是图像获取，即捕捉图像。2）接下来是预处理，它增强图像质量，并包括二值化以将内容与背景分离。3）接下来是布局分析，即将文档分割成不同区域。4）下一步是字符级分割，将文本分解为行、单词和单独字符。5）接着是识别，包括特征提取和分类以识别字符。6）最后，后处理通过经常使用语言模型来改进结果。所有这些阶段对于有效的 OCR 性能都是必不可少的（Jain et al., 2021; Nazeem et al., 2024）。

虽然光学字符识别（OCR）系统已经有了显著的进步，特别是在资源丰富的语言（HRL）如英语和法语方面（Nazeem et al., 2024），但对于复杂或低质量图像、历史文档和资源匮乏的语言（LRL）的文本识别仍然存在挑战（Agarwal and Anastasopoulos, 2024）。在本研究中，我们对各种多语言和单一语言的 OCR 系统进行了彻底的检验，评估了它们对南亚的两种选择的资源匮乏语言——僧伽罗语和泰米尔语的处理能力。

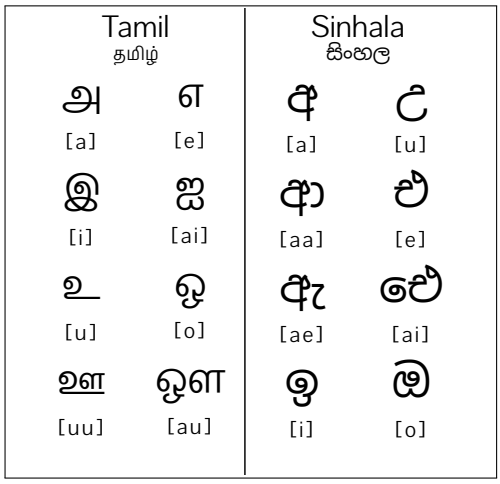



Figure 1: 泰米尔语和僧伽罗语中圆体字的一个使用示例。

僧伽罗语是一种印欧语言，大约只有 1600 万人作为第一语言使用，主要分布在斯里兰卡岛上（de Silva, 2025）。僧伽罗语有一种独特的文字，源自印度婆罗米文（Fernando, 1949）。泰米尔语是一种达罗毗荼语，大约有 7900 万人作为第一语言使用，主要分布在印度、斯里兰卡和新加坡（Wijeratne et al., 2019）。它也有一种独特的文字，也是印度婆罗米文的后

[1] https://huggingface.co/datasets/Nevidu/tamil_synthetic_ocr

裔（Paneerselvam, 1972）。根据 Ranathunga and de Silva (2022) 提出的标准，僧伽罗语和泰米尔语都被认为是低资源语言，其中僧伽罗语被认为是资源更少的语言（类别 02），而泰米尔语则是类别 03。

## 2 现有工作

尽管在过去几十年里进行了广泛的研究，识别光学字符识别（OCR）系统中的僧伽罗字符的挑战仍然是一个巨大的障碍（Anuradha et al., 2020）。如泰米尔语这样的印度语言，表现出众多的复杂性和字符变体，极大地增加了开发有效 OCR 解决方案的难度。明确来说，南亚圆形字母的准确性显著落后于拉丁系字母，这突显了一个关键的改进和发展领域（Anuradha et al., 2021）。

### 2.1 僧伽罗文 OCR 系统

已经进行了一些研究来开发僧伽罗语 OCR 系统。通过 Anuradha et al. (2020) 提出了一种使用带有图形用户界面的 Tesseract 4.0 OCR 引擎的僧伽罗语系统。该系统由五个主要组件组成：用户、API、Tesseract 引擎（Smith, 2007）、后处理器和数据存储。Tesseract 引擎使用基于 LSTM 的深度学习技术来处理图像并识别文本。然而，Tesseract 引擎无法识别某些字符。为了解决这个问题，后处理器识别出那些未识别的字符并应用语言规则以确保准确的输出。在这项研究中，使用了商用字体类型，并在大小上有所变化，结果平均准确率为 94 %。

一项关于多风格印刷僧伽罗字符识别的研究（Maduranga and Jayalal, 2022）使用了一种混合人工神经网络（ANN）模型，结合了以往研究的概念。该过程包括四个步骤：数据预处理用于图像增强和去噪；特征提取，将 50x50 像素的字符图像划分为 9 个区域，每个区域 12 个部分，以创建包含 108 个信号的特征向量供 ANN 使用；开发和训练，利用来自 850 字符数据库的线特征（主要是 Iskoola Pota 字体 [2]）在 MATLAB 中训练一个具有 108 个输入节点、78 个隐藏节点和 34 个输出节点的反向传播网络，经过 138 个周期后，达到大约 75 % 的训练准确率；以及测试，使用一个 1253 个字符的独立数据集进行性能评估。

Velayuthan and Ambegoda (2025) 进行了一项比较分析，通过对比多种知名的 OCR 模型来解决低资源语言文档数字化的挑战，这些模型包括：Surya [3]、TR-OCR [4]、EasyOCR [5] 和 Tesseract OCR（Smith, 2007）。虽然起初声称专注于僧伽罗语和泰米尔语，但该研究最终仅对僧伽罗语进行了实验，并使用了两个合成僧伽罗语数据集以及使用 FUNSD 数据集（Jaume et al., 2019）对英语进行了实验。通过采用如 CER 和 WER 之类的度量标准，评估过程包括数据过滤和后处理技术。研究结果显示，Surya 在僧伽罗语数据集上的表现明显优于其他模型。尽管所有模型在英语数据集上的错误率都很高，但 Surya 被认为是僧伽罗语的最佳选择，表现出相对较高的准确率，同时与 TR-OCR 相比，保持了适中的计算需求和卓越的能效。僧伽罗语和英语精度差异的很大一部分可以归因于数据集的性质，因为僧伽罗语数据集是合成生成的，而英语数据集则包含了形式文件的噪声图像。

### 2.2 泰米尔文 OCR 系统

已经开展了多项研究计划以创建用于泰米尔语的 OCR 系统。与主要在斯里兰卡使用的僧伽罗语不同，泰米尔语在南亚地区具有更广泛的地理分布。尽管它们有不同的语言根源，由于两者的书写系统相关，如图 1 所示，泰米尔语也使用类似于僧伽罗语的圆形文字，这为 OCR 技术的发展带来了类似的挑战。

Liyanage et al. (2015) 使用开源的 Tesseract OCR 引擎开发了一个泰米尔语 OCR 系统，灵感来源于其在僧伽罗语和孟加拉语等文字上的应用。该方法涉及创建一个包含 169 个字符的 OCR 字母表，并从各种 Unicode 字体中选取的单词准备训练数据。研究人员测试了不同的训练组合，发现使用来自三种字体的三种大小数据的模型取得了最佳效果。该系统在 20 张来自古泰米尔书籍的扫描图像上进行了评估，准确率达到了 81 %，比 Tesseract 现有的泰米尔模块提高了 12.5 %。

近期研究介绍了 Nayana 框架（Kolavi et al., 2025），该框架增强了如 GOT OCR（Wei et al., 2024）这样的视觉-语言模型（VLMs）在包括泰米尔语在内的低资源语言中的表现。它通过一个布局感知的合成数据生成流程和低秩适配（LoRA）来解决数据稀缺问题。该系统在保留布局的同时将英语文档翻译成泰米尔语，紧接着用 LoRA 进行两阶段的跨模态对齐训练。Nayana-OCR 在性能上取得了显著提升，字错误率（WER）为 0.551，METEOR 分数为 0.592，显著超越了基础的 GOT OCR 模型（WER 1.020，ME-

[2] https://learn.microsoft.com/en-us/typography/font-list/iskoola-pota

[3] https://github.com/VikParuchuri/surya

[4] https://huggingface.co/Ransaka/TrOCR-Sinhala

[5] https://github.com/JaidedAI/EasyOCR

TEOR 0.051）以及其他传统 OCR 系统，包括 Tesseract （Smith, 2007）和 PaddleOCR [6]，在泰米尔语测试集上的表现。

## 3 方法论

如前所述，研究人员正在利用开源工具对模型进行有效微调，以适应新语言并提升现有能力。许多这些工具提供多语言支持。此外，一些组织开发了在 OCR 方面表现出色的商业引擎。在本研究中，我们在零样本环境中对精选的 OCR 引擎在僧伽罗语和泰米尔语的能力进行了深入评估。

### 3.1 精选 OCR 技术概述

在本研究中，我们评估了六种专门为 OCR 任务设计的开源和商业引擎的能力。

Cloud Vision API [7]：Cloud Vision API 使开发者能够无缝地将视觉检测功能融入他们的应用程序。这包括图像标注、面部和地标检测、OCR，以及对敏感内容的标记等功能。API 的第一个版本于 2017 年 5 月正式推出。该 API 被设计用于对 PDF 和 TIFF 文件以及密集文本的图像进行 OCR 处理。它特别针对包含大量文本和手写内容的图像文档进行了优化，以实现准确的识别和转换为机器可读文本。

文档 AI [8]：文档 AI 是一个文档理解平台，可以将文档中的非结构化数据转化为结构化数据，使其更易于理解、分析和利用。它采用机器学习和谷歌云来开发可扩展的、端到端的基于云的文档处理应用程序。API 提供通过内容分类、实体抽取、高级搜索等方式进行组织。OCR 处理器特别支持从文档中识别和提取文本，包括手写文本，支持超过 200 种语言。此外，处理器还使用机器学习根据内容的可读性来评估文档的质量。

Tesseract：最著名和最常用的 OCR 引擎之一是 Tesseract OCR。这个开源项目最初由 HP 开发，现在由 Google 资助，提供了优秀的文本识别能力。Tesseract 结合了隐藏马尔可夫模型（HMMs）和多种机器学习算法，与传统的计算机视觉技术一起用于识别文本。Tesseract 4.0 引入了深度学习方法，与早期主要依赖传统方法的版本相比，显著提高了性能。Tesseract 使用的深度学习模型基于长短期记忆（LSTM）网络（Smith, 2007; Nazeem et al., 2024）。在本研究中，我们使用了 Tesseract 5.5.0，这是 Tesseract 4.0 的改进版本，结合了现有的 LSTM 引擎，并有几项性能改进。

Subasa OCR [9]：Anuradha et al. (2020) 的研究进行了扩展，其中涉及了一系列基于深度学习（LSTM）的 Tesseract 4.0 实验，以通过研究文本体裁、图像分辨率和算法复杂性来估计僧伽罗文 OCR 的复杂性。训练数据主要来自 UCSC 10M 僧伽罗文语料库 [10]，使用了各种僧伽罗文字体和图像质量，涉及字符分割和迭代训练过程。评估在 30 张不同的测试图像上进行，这些图像被分类为旧报纸（200 DPI）、旧书（72 DPI）和当代书籍（300 DPI），并对低 DPI（96px）的当代图像进行了额外测试。准确性是主要指标，其通过比较原始文本和 OCR 输出的字符计数来计算。Tesseract 4.0 在旧报纸上实现了高达 67.02 % 的字符准确率，在旧书上高达 87.53 %，在当代书籍上高达 87.63 %，甚至在低 DPI 的当代图像上保持了高达 87.88 % 的高准确率 (Anuradha et al., 2021)。

Surya [3]：这是一个 OCR 工具包，支持超过 90 种语言，并在与云服务的对比中表现良好。它具有任意语言的行级文本检测功能、版面分析（包括表格、图像、标题等的检测）、阅读顺序检测和表格识别（检测行和列），以及 LaTeX OCR 能力。该文本检测模型使用四个 A6000 GPU 训练了三天，并使用多样化的图像集以构建系统。该模型基于一个为语义分割修改的 EfficientViT 架构 (Liu et al., 2023) 从头开始设计。同时，文本识别模型在相同硬件上训练了两周，采用了一个修改后的 Donut 模型 (Kim et al., 2022)，该模型结合了分组查询注意力（GQA） (Ainslie et al., 2023)、专家混合（MoE）层 (Shazeer et al.)、UTF-16 解码以及层配置的更改。需要注意的是，该系统设计用于印刷文本而非手写体。

EasyOCR [5]：这是一种支持 80 多种语言的 OCR 技术。EasyOCR 利用 ResNet (He et al., 2016)、LSTM 和 CTC（连接时序分类） (Graves et al., 2006) 模型进行字符识别。EasyOCR 的检测组件使用 CRAFT 算法 (Baek et al., 2019)。EasyOCR 由三个关键要素组成。第一个是特征提取，由 ResNet 模型执行。第二个要素是序列标注，其中使用了 LSTM 算法，最后一个组件是解码。解码依赖于 CTC。EasyOCR 的 Readtext 函数在识别过程中使用。EasyOCR 的一个显著特征是能够从图像中读取字母和数字，并提供它们位置的坐标 (Awalgaonkar et al., 2021)。

Google Cloud Vision API 和 Document AI 是商业引擎，而 Google Tesseract、Surya 和 EasyOCR 是可以微调的开源系统。此外，我

[6] https://github.com/PaddlePaddle/PaddleOCR
[7] https://cloud.google.com/vision/docs/ocr
[8] https://cloud.google.com/document-ai/
[9] https://ocr.subasa.lk/
[10] https://ltrl.ucsc.lk/tools-and-resourses/

们选择了 Subasa OCR，这是一种通过网络应用程序 [9] 提供的经过微调的 Tesseract 模型，尽管源代码和模型不能直接访问。

### 3.2 数据集选择与组装

为了实现最佳效果，我们为所选择的两种语言分别使用不同的数据集，通过这种量身定制的方法来提高我们分析的有效性。对于僧伽罗语，我们选择了一个在 Hugging Face 上发布的数据集，该数据集由 Ravihara (2024) 提供，包括 6,969 对图像和参考文本。

由于我们考虑了一个用于僧伽罗语的合成生成数据集，我们还旨在评估用于泰米尔语的合成生成数据，以确保公平比较。然而，我们找不到任何以类似方式开发的公开可用的泰米尔语数据集。因此，我们决定为泰米尔语创建一个新数据集。泰米尔语数据集创建的概述如图 2 所示。

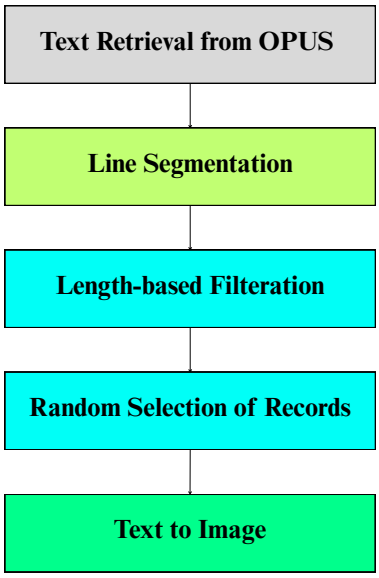



Figure 2: 泰米尔合成 OCR 数据集创建概述。

泰米尔文本是从 OPUS [11] 获取的，特别选择了 OpenSubtitles v2024 [12] 。内容随后按行划分，只关注泰米尔字符，因为主要目标是语言评估，最终得到 2,437,960 条记录。接下来，这个集合被过滤，只保留了超过 40 个字符的文本，得到 222,658 条记录。这一步过滤确保了词级评估的准确性。然而，为了与僧伽罗数据集进行公平比较，我们决定从剩余文本中随机选择 7,000 个样本以平衡样本量。不管怎样，由于所需资源的消耗，进行更多样本的 OCR 是具有挑战性的。

[11] https://opus.nlpl.eu/

[12] https://opus.nlpl.eu/OpenSubtitles/ta&en/v2024/OpenSubtitles

我们精心挑选了六种独特的字体，以多样化从文本记录生成图像的过程，确保每种视觉表现形式在分析中都具有影响力。选定的字体如下所示：

- 辛德马杜赖 [13]
- Noto Serif Tamil [14]
- 卡维瓦纳尔 [15]
- Noto Sans Tamil [16]
- Pavanam [17]
- 多谢泰米尔 [18]

然后开发了一个函数，利用 Pillow 库的功能系统地将文本数据转换为图像文件。其基本目的是确保输入文本记录在定义的一组字体文件中按比例分布，以促进在生成的图像数据集中公平使用字体。对于每个文本条目，该函数会根据文本边框的测量值动态计算最佳图像尺寸，将文本以黑色渲染在白色背景上。此实现的一个显著特点是精确地将文本居中于生成的图像内，这是通过相对于图像的高度和宽度计算定位来实现的，从而增强了视觉的一致性和质量。

对于泰米尔语数据集的后处理阶段，涉及排除那些在参考和生成特征中均不包含字符串值的记录。相比之下，僧伽罗语数据集的后处理包含了一个额外步骤，旨在从参考和生成文本中消除所有非僧伽罗字符。此步骤的实施是为了确保评估集中在系统的语言能力上。由于我们创建了泰米尔语数据集，因此专注于语言字符的后处理步骤在预处理时已解决。这个新颖的合成泰米尔语 OCR 公共数据集是我们在这项研究中的贡献之一 [1] 。我们合成数据集中一些泰米尔语句子的示例如图 3 所示。

### 3.3 OCR 系统的集成

Google Cloud Vision API 和 Document AI 都可以通过 Google Cloud Platform (GCP) 使用。然而，启用 Cloud Vision API 很简单，因为它只需要在 GCP 中激活该 API，而 Document AI 需要在 GCP 中手动创建一个处理器。这个创建的处理器随后用于启动 OCR 过程。

Tesseract 引擎的集成过程通过 pytesseract 库变得相当简单。这个库使得每个数据集中的

[13] https://fonts.google.com/specimen/Hind+Madurai

[14] https://fonts.google.com/noto/specimen/Noto+Serif+Tamil

[15] https://fonts.google.com/specimen/Kavivanar

[16] https://fonts.google.com/noto/specimen/Noto+Sans+Tamil

[17] https://fonts.google.com/specimen/Pavanam

[18] https://fonts.google.com/specimen/Anek+Tamil


Figure 3: 来自我们的数据集中，分别取自《Hind Madurai》、《Anek Tamil》和《Kavinar》的三个泰米尔语句子示例。

每个记录的字符识别自动化，从而简化了整个数据处理工作流程。

如前所述，Subasa OCR 引擎仅能通过网络应用程序进行访问。由于这一限制，我们必须手动逐一输入图片来执行 OCR，这对于我们庞大的 6,969 条记录的僧伽罗语数据集而言，显得非常不切实际。为了简化这一繁琐的任务并提高效率，我们决定使用 Selenium 自动化此过程，经过仔细检查网页源代码来识别元素位置后进行。这个战略转变不仅减轻了手动工作量，而且大大缩短了处理时间。

Surya 和 EasyOCR 的无缝集成是非常简单的，这主要归功于它们优秀的文档。这两个引擎都可以在 GitHub 上访问，并且可以直接通过 Python 方便地安装为库，使得安装过程高效且用户友好。

### 3.4 评估机制

评估是通过将生成的文本与数据集的参考文本进行比较来完成的。为了进行比较，我们使用了五种不同的度量方法。它们列举如下。

字符错误率 (CER)：它基于 Levenshtein 距离的概念，测量将真实文本转换为 OCR 生成的输出所需的最小字符级操作数量（替换、删除和插入）。CER 公式表示为 $(S+I+D)/N$ ，其中 $S$ 表示替换的数量，$I$ 表示插入的数量，$D$ 表示删除的数量，$N$ 表示真实文本中的字符总数 (Nazeem et al., 2024) 。

字错误率 (WER)：与 CER 类似，WER 通过将 OCR 系统生成的文本与真实或参考文本进行比较来计算。WER 由 OCR 引擎产生的字级错误数量决定。计算字错误率的公式也是 $(S+I+D)/N$ ，但考虑的是字级而不是字符级 (Nazeem et al., 2024) 。

双语评估学习 (BLEU): 这是一种用于自动评估机器翻译质量的方法。其基本原理是，机器生成的翻译与一个或多个专业人工翻译越接近，其质量就越高。BLEU 通过一个依赖于高质量人工参考译文的数值指标来评估这种接近程度。该方法包括针对这些参考译文的可变长度短语匹配的加权平均值，使用称为修正的 n-gram 精度的概念。此外，它还引入了简短惩罚，以避免候选译文在相对于参考译文过于简短时的情况。最终的 BLEU 得分范围从 0 到 1，通过计算修正的 n-gram 精度的几何平均值并乘以简短惩罚获得 (Papineni et al., 2002) 。

平均归一化 Levenshtein 相似度（ANLS）：此指标同时考虑推理错误和 OCR 的缺陷。为了评估答案，它使用 Levenshtein 距离在模型的响应与真实值之间计算相似度得分。该评分系统的一个关键特征是其在归一化 Levenshtein 距离（NL）上的阈值应用（$\tau$ =0.5）：如果 NL 小于或等于 0.5，则相似度得分计算为 1-NL；否则，得分为 0。这种方法允许 ANLS 为逻辑上正确但可能包含轻微识别错误的响应提供中间得分（范围从 0.5 到 1），与标准准确性指标相对，它会将其得分为零 (Biten et al., 2019) 。

$$\text{ANLS} = \frac{1}{N}\sum_{i=0}^{N}\left(\max_{j} s(a_{ij}, o_{q_i})\right) \quad (1)$$

$$s(a_{ij}, o_{q_i}) = \begin{cases} 1 - NL(a_{ij}, o_{q_i}) & \text{if } NL(a_{ij}, o_{q_i}) < \tau \\ 0 & \text{if } NL(a_{ij}, o_{q_i}) \geq \tau \end{cases}$$

评估翻译准确顺序的度量 (METEOR)：METEOR 是一种用于评估机器翻译质量的指标。它通过识别机器生成的输出与人工参考翻译之间的单元匹配来运作，允许基于表面形式、词干形式和同义词进行匹配。专门设计用于解决 BLEU 指标的局限性，例如其无法直接考虑召回率以及对词序的间接测量，METEOR 通过结合单元精确度、单元召回率（更强调召回率）和评估匹配词序的碎片惩罚来计算分数。

与其他评估方法相比，这种方法已展示出与人工判断更好的相关性。鉴于其增强的能力能够评估生成文本与参考文本的匹配程度，并且在 OCR 研究中迅速被采用，我们选择使用这个指标（Banerjee and Lavie, 2005）。

### 3.5 结果讨论

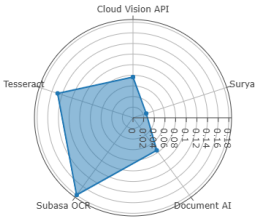
在本节中，我们展示了从 OCR 系统比较评估中得出的结果分析。研究结果在表格 1 中分别以僧伽罗语和泰米尔语展示。除了 Subasa OCR 和 EasyOCR，其余四个系统都能够处理本研究中选定用于评估的两种语言。需要注意的是，Subasa OCR 是一个专门为僧伽罗语调整的单语系统。虽然 EasyOCR 具备多语言能力，但显然不支持僧伽罗语。

在对两种语言的整体结果进行评估时，Cloud Vision API 和 Document AI 取得了非常相似的结果。值得注意的是，Document AI 在所有指标上都比其他引擎在泰米尔语中表现得更好。

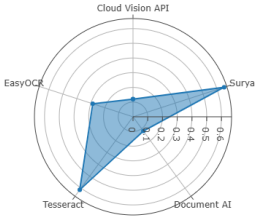
总体而言，泰米尔语的最佳 WER 结果明显高于僧伽罗语。这一观察结果突显了字符和单词识别之间的差异。尽管 Document AI 实现了非常低的 CER，但相对较高的 WER 表明系统虽然能够有效识别字符，却在泰米尔语言的单词形成和间距上存在困难。相比之下，两种语言的 METEOR 和 ANLS 分数都相对较高，表明在内容、词序和语义意义上有较强的一致性。然而，泰米尔语的 BLEU 分数明显低于其他指标，可能是由于较高的 WER，导致较少成功的 n 元词组重叠。

Surya 在僧伽罗语上的表现堪称非凡，成为其中的亮点。指标清楚地展示了这一成功，如表 1 所示。当我们比较泰米尔语的最佳 WER 为 11.98 % 与僧伽罗语令人印象深刻的 2.61 % 时，如图 4 所示，Surya 引擎在僧伽罗语上的准确性优越性变得异常明显。此外，METEOR 和 ANLS 得分分别为 0.9723 和 0.9920，进一步强调了其近乎完美的词级表现。这些数据强有力地展示了 Surya 在处理僧伽罗语方面的卓越能力。

Subasa OCR 和 Tesseract 之间的比较尤其具有吸引力，因为 Subasa OCR 代表了对 Tesseract 4.0 引擎的针对性修改，专门用于僧伽罗语。Subasa 的作者声称，他们的修改产生了显著优于标准 Tesseract 4.0（Anuradha et al., 2021）的结果。度量评估显示，Tesseract 5.5.0 在所有指标上都优于 Subasa OCR。这表明谷歌最新版本的 Tesseract 在其原版中，甚至对僧伽罗语进行了实质的增强。然而，Tesseract 在泰米尔语中的表现与其他系统相比并不具竞争力。如前所述，由于 EasyOCR 不支持僧伽罗语，因此仅在泰米尔语上进行了评估，而在我们比较的开源系统中，它表现出了最优越的性能。虽然与两个商业引擎相比，结果显示出显著下降，但与其他开源解决方案的对比却十分显著。




(a) 僧伽罗语




(b) 泰米尔语

Figure 4: 僧伽罗语和泰米尔语的 WER 结果

## 4 结论

在本研究中，我们在零样本设置下评估了用于两种不同南亚语言的六种 OCR 引擎。为了便于评估，我们创建了一个合成的泰米尔语 OCR 数据集，使用六种不同的字体，以与现有的僧伽罗语数据集平行。所选 OCR 系统的性能通过五种测量方法进行全面分析，这些方法评估了字符和单词级别的精确度。

结果表明，Document AI 在泰米尔语上的表现最佳，而 Surya 在僧伽罗语上表现出色。云视觉 API 和 Document AI 在僧伽罗语和泰米尔语上的总体表现是合理的，突出了商业引擎在 OCR 领域的能力，如预期的那样。一个特别出色的表现者是 Surya 在僧伽罗语上的表现，在每个指标上都超过了所有其他 OCR 系统。此外，泰米尔语中最佳 CER 和 WER 结果之间的显著差异表明，虽然系统在字符识别上表现出色，但在通过正确字符形成和空白检测准确识别单词方面有所欠缺。此外，值得注意的是，Zero-Shot Tesseract 5.5.0 的表现优于针对僧伽罗语（Subasa OCR）微调的 Tesseract 4.0 系统。

| OCR System | Language | CER ↓ | WER ↓ | BLEU ↑ | ANLS ↑ | METEOR ↑ |
|---|---|---|---|---|---|---|
| Cloud Vision API | Sinhala | 0.0619 | 0.0767 | 0.9193 | 0.9447 | 0.9269 |
| | Tamil | 0.0079 | 0.1204 | 0.5790 | 0.9922 | 0.8751 |
| Surya | Sinhala | 0.0076 | 0.0261 | 0.9396 | 0.9920 | 0.9723 |
| | Tamil | 0.1392 | 0.64999 | 0.1487 | 0.8672 | 0.3359 |
| Document AI | Sinhala | 0.0610 | 0.0758 | 0.9199 | 0.9455 | 0.9278 |
| | Tamil | 0.0078 | 0.1198 | 0.5803 | 0.9923 | 0.8762 |
| Subasa OCR | Sinhala | 0.0761 | 0.1799 | 0.6894 | 0.9259 | 0.8099 |
| | Tamil | - | - | - | - | - |
| Tesseract | Sinhala | 0.0702 | 0.1489 | 0.7553 | 0.9319 | 0.8436 |
| | Tamil | 0.0780 | 0.6145 | 0.0493 | 0.9264 | 0.3201 |
| EasyOCR | Sinhala | - | - | - | - | - |
| | Tamil | 0.1172 | 0.2876 | 0.3461 | 0.8828 | 0.6744 |

Table 1: 僧伽罗语和泰米尔语的 OCR 系统评估

5

局限性

分析集中于比较使用泰米尔语和僧伽罗语印刷文本的选定 OCR 引擎的性能。然而，需要注意的是所使用数据集的一个局限性；两者都是合成生成的，以白色背景上的黑色文本为特征。这种设计产生了干净和清晰的图像，但无法准确反映捕获印刷文本时的现实世界情况。OCR 技术面对与输入图像质量相关的重大挑战，尤其是在处理历史文献或低资源语言数据时。诸如印刷质量差、低分辨率、阴影、模糊、透光效果、污渍和倾斜等因素可以严重影响 OCR 的准确性。具有失真、纹理背景、杂乱环境、断开的线段、分散的点、断行、旋转、运动模糊和失焦模糊的图像会使字符分割和识别复杂化，通常导致更高的错误率。此外，低图像分辨率可能会妨碍整体的 OCR 速度，因为字符表示的不确定性会导致更多识别变体。因此，当使用相机拍摄的图像进行评估时，准确性水平可能会显著不同于本研究中呈现的结果。

## References


Milind Agarwal and Antonios Anastasopoulos. 2024. A concise survey of OCR for low-resource languages. In Proceedings of the 4th Workshop on Natural Language Processing for Indigenous Languages of the Americas (AmericasNLP 2024), pages 88–102, Mexico City, Mexico. Association for Computational Linguistics.

Joshua Ainslie, James Lee-Thorp, Michiel de Jong, Yury Zemlyanskiy, Federico Lebron, and Sumit Sanghai. 2023. Gqa: Training generalized multi-query transformer models from multi-head checkpoints. In Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, pages 4895–4901.

Isuri Anuradha, Chamila Liyanage, and Ruvan Weerasinghe. 2021. Estimating the Effects of Text Genre, Image Resolution and Algorithmic Complexity needed for Sinhala Optical Character Recognition. International Journal on Advances in ICT for Emerging Regions (ICTer), 14(3).

Isuri Anuradha, Chamila Liyanage, Harsha Wijayawardhana, and Ruvan Weerasinghe. 2020. Deep Learning Based Sinhala Optical Character Recognition (OCR). In 2020 20th International Conference on Advances in ICT for Emerging Regions (ICTer), pages 298–299. IEEE.

Ninad Awalgaonkar, Prashant Bartakke, and Ravindra Chaugule. 2021. Automatic License Plate Recognition System Using SSD. In 2021 international symposium of Asian control association on intelligent robotics and industrial automation (IRIA), pages 394–399. IEEE.

Youngmin Baek, Bado Lee, Dongyoon Han, Sangdoo Yun, and Hwalsuk Lee. 2019. Character region awareness for text detection. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pages 9365–9374.

Satanjeev Banerjee and Alon Lavie. 2005. METEOR: An automatic metric for MT evaluation with improved correlation with human judgments. In Proceedings of the ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization, pages 65–72, Ann Arbor, Michigan. Association for Computational Linguistics.

Ali Furkan Biten, Ruben Tito, Andres Mafla, Lluis Gomez, Marçal Rusinol, Ernest Valveny, CV Jawahar, and Dimosthenis Karatzas. 2019.

Scene Text Visual Question Answering. In Proceedings of the IEEE/CVF international conference on computer vision, pages 4291–4301.

PEE Fernando. 1949. Palaeographical Development of the Brahmi Script in Ceylon from 3rd Century BC to 7th Century AD.

Alex Graves, Santiago Fernández, Faustino Gomez, and Jürgen Schmidhuber. 2006. Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks. In Proceedings of the 23rd international conference on Machine learning, pages 369–376.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 770–778.

Pooja Jain, Kavita Taneja, and Harmunish Taneja. 2021. Which OCR toolset is good and why: A comparative study. Kuwait Journal of Science, 48(2).

Guillaume Jaume, Hazim Kemal Ekenel, and Jean-Philippe Thiran. 2019. Funsd: A dataset for form understanding in noisy scanned documents. In 2019 International Conference on Document Analysis and Recognition Workshops (ICDARW), volume 2, pages 1–6. IEEE.

Geewook Kim, Teakgyu Hong, Moonbin Yim, JeongYeon Nam, Jinyoung Park, Jinyeong Yim, Wonseok Hwang, Sangdoo Yun, Dongyoon Han, and Seunghyun Park. 2022. Ocr-free document understanding transformer. In European Conference on Computer Vision, pages 498–517. Springer.

Adithya Kolavi, Samarth P, and Vyoman Jain. 2025. Nayana OCR: A scalable framework for document OCR in low-resource languages. In Proceedings of the 1st Workshop on Language Models for Underserved Communities (LM4UC 2025), pages 86–103, Albuquerque, New Mexico. Association for Computational Linguistics.

Xinyu Liu, Houwen Peng, Ningxin Zheng, Yuqing Yang, Han Hu, and Yixuan Yuan. 2023. Efficientvit: Memory efficient vision transformer with cascaded group attention. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pages 14420–14430.

Chamila Liyanage, Thilini Nadungodage, and Ruvan Weerasinghe. 2015. Developing a commercial grade Tamil OCR for recognizing font and size independent text. In 2015 Fifteenth International Conference on Advances in ICT for Emerging Regions (ICTer), pages 130–134. IEEE.

YVANT Maduranga and Shantha Jayalal. 2022. Multi-Style Printed Sinhala Character Recognition and Digitalization Using Artificial Neural Network. In 2022 2nd International Conference on Advanced Research in Computing (ICARC), pages 120–124. IEEE.

Rishabh Mittal and Anchal Garg. 2020. Text extraction using OCR: A Systematic Review. In 2020 second international conference on inventive research in computing applications (ICIRCA), pages 357–362. IEEE.

Meharuniza Nazeem, Anitha R, Navaneeth S, and Rajeev R. R. 2024. Open-source OCR libraries: A comprehensive study for low resource language. In Proceedings of the 21st International Conference on Natural Language Processing (ICON), pages 416–421, AU-KBC Research Centre, Chennai, India. NLP Association of India (NLPAI).

R Paneerselvam. 1972. A critical study of the Tamil Brahmi insciptions. Acta Orientalia, 34:35–35.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics, pages 311–318, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics.

Surangika Ranathunga and Nisansa de Silva. 2022. Some languages are more equal than others: Probing deeper into the linguistic disparity in the NLP world. In Proceedings of the 2nd Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 12th International Joint Conference on Natural Language Processing (Volume 1: Long Papers), pages 823–848, Online only. Association for Computational Linguistics.

Ransaka Ravihara. 2024. sinhala_synthetic_ocr-large (revision f3cac3b).

Noam Shazeer, Azalia Mirhoseini, Krzysztof Maziarz, Andy Davis, Quoc Le, and Jeff Dean. Outrageously large neural networks: The sparsely-gated mixture-of-experts layer.

Nisansa de Silva. 2025. Survey on Publicly Available Sinhala Natural Language Processing Tools and Research. arXiv preprint arXiv:1906.02358v25.

Ray Smith. 2007. An overview of the Tesseract OCR engine. In Ninth international conference on document analysis and recognition (ICDAR 2007), volume 2, pages 629–633. IEEE.

Purushoth Velayuthan and Thanuja Ambegoda. 2025. Benchmarking ocr models for sinhala and tamil document digitization.

Ruvan Weerasinghe, Asanka Wasala, Dulip Herath, and Viraj Welgama. 2008. NLP applications of Sinhala: TTS & OCR. In Proceedings of the Third International Joint Conference on Natural Language Processing: Volume-II.

Haoran Wei, Chenglong Liu, Jinyue Chen, Jia Wang, Lingyu Kong, Yanming Xu, Zheng Ge, Liang Zhao, Jianjian Sun, Yuang Peng, et al. 2024. General OCR Theory: Towards OCR-2.0 via a Unified End-to-end Model.

Yudhanjaya Wijeratne, Nisansa de Silva, and Yashothara Shanmugarajah. 2019. Natural Language Processing for Government: Problems and Potential. International Development Research Centre (Canada), 1.