ICS 35.080
L 77

AIOSS

中国人工智能开源软件发展联盟标准

AIOSS—01—2018

人工智能　深度学习算法评估规范

Artificial intelligence—Assessment specification for deep learning algorithms

2018－07－01 发布　　2018－07－01 实施

目　　次

前　言

本标准按照GB/T 1.1—2009给出的规则起草。

请注意本文件的某些内容可能涉及专利。本文件的发布机构不承担识别这些专利的责任。

本标准由中国人工智能开源软件发展联盟提出。

本标准由中国电子技术标准化研究院归口。

本标准负责起草单位：中国电子技术标准化研究院、中国科学院软件研究所、上海计算机软件技术开发中心、北京航空航天大学、华东师范大学、中国科学院计算技术研究所、军事科学院国防科技创新研究院、国防科技大学、卡索(北京)科技有限公司、北京百度网讯科技有限公司、浙江蚂蚁小微金融服务集团有限公司、深圳前海微众银行股份有限公司、顺丰科技有限公司、深圳市优必选科技有限公司、北京京东尚科信息技术有限公司、深圳赛西信息技术有限公司、数据地平线（广州）科技有限公司。

本标准主要起草人：薛云志、孟令中、崔静、张明英、张璨、周平、武斌、郭崎、刘畅、吴涛、李海峰、肖良、张超、于泉杰、宋俊典、戴炳荣、王长波、孙仕亮、陈美、李刚、潘欣、程思、刘志欣、刘新凯、王太峰、巢林林、袁杰、曹安然、尹思遥。

引　　言

人工智能的迅速发展正在深刻改变人类社会生活、改变世界，其技术和应用正经历快速发展的阶段。根据GB/T 5271.28-2001《信息技术 词汇 第28部分:人工智能 基本概念与专家系统》中的定义，“人工智能是表现出与人类智能（如推理和学习）相关的各种功能单元的能力。”机器学习是人工智能的核心技术之一，是使计算机具有智能的重要途径，其应用遍及人工智能的各个领域。深度学习是机器学习的一个子集，发源于人工神经网络的研究，通常也称为深度神经网络，是一种基于数据进行表征学习的方法。目前，深度学习算法在金融、安防、医疗等领域得到广泛应用，国务院发布的《新一代人工智能发展规划》中指出，人工智能进入新的发展阶段，“呈现出深度学习、跨界融合、人机协同、群智开放、自主操控等新特征。”

然而，业界缺乏对深度学习算法可靠性、可移植性、效率等的系统性评估方法，一定程度上影响着深度学习的广泛应用和技术发展。本标准此版本仅针对人工智能深度学习算法的可靠性评估进行要求。随着研究的深入及应用的发展，后续将不断进行持续改进，逐渐扩展到深度学习算法可移植性、效率等方面的评估。

中国电子技术标准化研究院作为国际标准化组织ISO/IEC JTC1/SC42（人工智能分技术委员会）的国内技术归口单位，在本标准研制过程中，充分发挥了组织协调和技术方向把关作用。标准编写组聚集了国内人工智能深度学习领域的技术专家，开展了多种形式的专题研讨和征求意见活动，在保证标准科学性、合理性和可行性的同时，也确保了标准研制过程的公开性和透明性。

本标准的研制工作，得到了中国人工智能开源软件发展联盟专家委员会的指导和支持，专家委员会主任委员、中国科学院院士、华东师范大学计算机科学与软件工程学院院长何积丰院士和各位专家对标准内容和文稿进行了深入严谨的讨论，给出了许多切实可行的意见，对标准质量提升和标准内容完善起到关键性作用。

使用帮助信息：任何单位和个人在使用本标准的过程中，若存在疑问，或有对本标准的改进建议和意见，请与中国电子技术标准化研究院（中国人工智能开源软件发展联盟 秘书处）联系。

电话：010-64102848；电子邮件：aioss@cesi.cn

通信地址：北京东城区安定门东大街1号（100007）

为了推动本标准的持续改进，使其内容更加贴近用户组织的实际需求，欢迎社会各方力量参加本标准的持续改进，本标准的更多信息欢迎关注“中国人工智能开源软件发展联盟”公众号。

人工智能　深度学习算法评估规范

1　范围

本标准提出了人工智能深度学习算法的评估指标体系、评估流程，以及需求阶段评估、设计阶段评估、实现阶段评估和运行阶段评估等内容。

本标准适用于指导深度学习算法开发方、用户方以及第三方等相关组织对深度学习算法的可靠性开展评估工作。

2　术语和定义

下列术语和定义适用于本文件。

2.1

可靠性　reliability

在规定的条件下和规定的时间内，深度学习算法正确完成预期功能，且不引起系统失效或异常的能力。

2.2

可靠性评估　reliability assessment

确定现有深度学习算法的可靠性所达到的预期水平的过程。

2.3

算法失效　algorithm failure

算法丧失完成规定功能的能力的事件。

2.4

危险　hazard

深度学习算法发生算法失效，从而导致机器学习系统出现的一个非预期或有害的行为，或者提交给其他与机器学习系统相关联的系统发生错误。

2.5

危险严重性　hazard severity

某种危险可能引起的事故后果的严重程度。

2.6

查准率　precision

对于给定的数据集，预测为正例的样本中真正例样本的比率。

2.7

查全率　recall

对于给定的数据集，预测为真正例的样本占所有实际为正例样本的比率。

2.8

准确率　accuracy

对于给定的数据集，正确分类的样本数占总样本数的比率。

2.9

响应时间　　response time

在给定的软硬件环境下，深度学习算法对给定的数据进行运算并获得结果所需要的时间。

2.10

对抗性样本　　adversarial examples

在数据集中通过故意添加细微的干扰所形成输入样本，受干扰之后的输入导致模型以高置信度给出错误的输出。

2.11

置信度　　confidence

总体参数值落在样本统计值某一区内的概率。

3　评估指标体系

3.1　评估指标体系表

基于深度学习算法可靠性的内外部影响考虑，结合用户实际的应用场景，本标准给出了一套深度学习算法的可靠性评估指标体系。本指标体系如图1所示，包含7个一级指标和20个二级指标。在实施评估过程中，应根据可靠性目标选取相应指标。

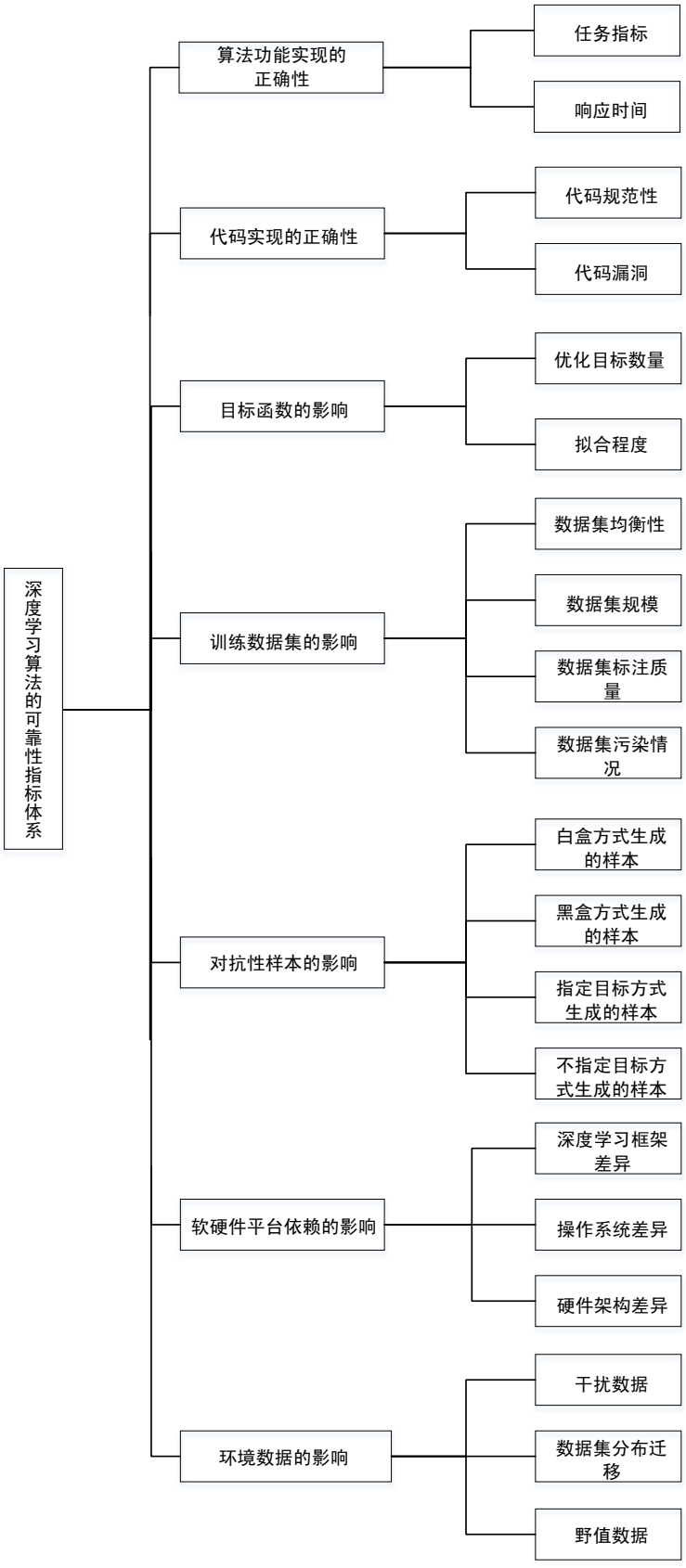

图 1　深度学习算法可靠性评估指标体系

3.2 算法功能实现的正确性

用于评估深度学习算法实现的功能是否满足要求，应包括但不限于下列内容：

a) 任务指标：用户可以根据实际的应用场景选择任务相关的基本指标，用于评估算法完成功能的能力；

示例：分类任务中的查准率（见 2.6）、查全率（见 2.7）、准确率（见 2.8）等；语音识别任务中的词错误率、句错误率等；目标检测任务中的平均正确率等；算法在使用中错误偏差程度带来的影响等。

b) 响应时间（见2.9）。

3.3 代码实现的正确性

用于评估代码实现功能的正确性，应包括下列内容：

a) 代码规范性：代码的声明定义、版面书写、指针使用、分支控制、跳转控制、运算处理、函数调用、语句使用、循环控制、类型转换、初始化、比较判断和变量使用等是否符合相关标准或规范中的编程要求；

b) 代码漏洞：指代码中是否存在漏洞。

示例：栈溢出漏洞、堆栈溢出漏洞、整数溢出、数组越界、缓冲区溢出等。

3.4 目标函数的影响

用于评估计算预测结果与真实结果之间的误差，应包括下列内容：

a) 优化目标数量：包括优化目标不足或过多。优化目标过少容易造成模型的适应性过强，优化目标过多容易造成模型收敛困难；

b) 拟合程度：包括过拟合或欠拟合。过拟合是指模型对训练数据过度适应，通常由于模型过度地学习训练数据中的细节和噪声，从而导致模型在训练数据上表现很好，而在测试数据上表现很差，也即模型的泛化性能变差。欠拟合是指模型对训练数据不能很好地拟合，通常由于模型过于简单造成，需要调整算法使得模型表达能力更强。

3.5 训练数据集的影响

用于评估训练数据集带来的影响，应包括下列内容：

a) 数据集均衡性：指数据集包含的各种类别的样本数量一致程度和数据集样本分布的偏差程度；

b) 数据集规模：通常用样本数量来衡量，大规模数据集通常具有更好的样本多样性；

c) 数据集标注质量：指数据集标注信息是否完备并准确无误；

d) 数据集污染情况：指数据集被人为添加的恶意数据的程度。

3.6 对抗性样本的影响

用于评估对抗性样本对深度学习算法的影响，应包括下列内容：

a) 白盒方式生成的样本：指目标模型已知的情况下，利用梯度下降等方式生成对抗性样本；

b) 黑盒方式生成的样本：指目标模型未知的情况下，利用一个替代模型进行模型估计，针对替代模型使用白盒方式生成对抗性样本；

c) 指定目标生成的样本：指利用已有数据集中的样本，通过指定样本的方式生成对抗性样本；

d) 不指定目标生成的样本：指利用已有数据集中的样本，通过不指定样本（或使用全部样本）的方式生成对抗性样本。

3.7 软硬件平台依赖的影响

用于评估运行深度学习算法的软硬件平台对可靠性的影响，应包括下列内容：

a) 深度学习框架差异：指不同的深度学习框架在其所支持的编程语言、模型设计、接口设计、分布式性能等方面的差异对深度学习算法可靠性的影响；
b) 操作系统差异：指操作系统的用户可操作性、设备独立性、可移植性、系统安全性等方面的差异对深度学习算法可靠性的影响；
c) 硬件架构差异：指不同的硬件架构及其计算能力、处理精度等方面的差异对深度学习算法可靠性的影响。

3.8 环境数据的影响

用于评估实际运行环境对算法的影响，应包括下列内容：

a) 干扰数据：指由于环境的复杂性所产生的非预期的真实数据，可能影响算法的可靠性；
b) 数据集分布迁移：算法通常假设训练数据样本和真实数据样本服从相同分布，但在算法实际使用中，数据集分布可能发生迁移，即真实数据集分布与训练数据集分布之间存在差异性；
c) 野值数据：指一些极端的观察值。在一组数据中可能有少数数据与其余的数据差别比较大，也称为异常观察值。

4 评估流程

4.1 概述

深度学习算法的可靠性评估流程如图 2所示。包括确定可靠性目标、选择评估指标、需求阶段的评估、设计阶段的评估、实现阶段的评估、运行阶段的评估及得出评估结论这七个活动。

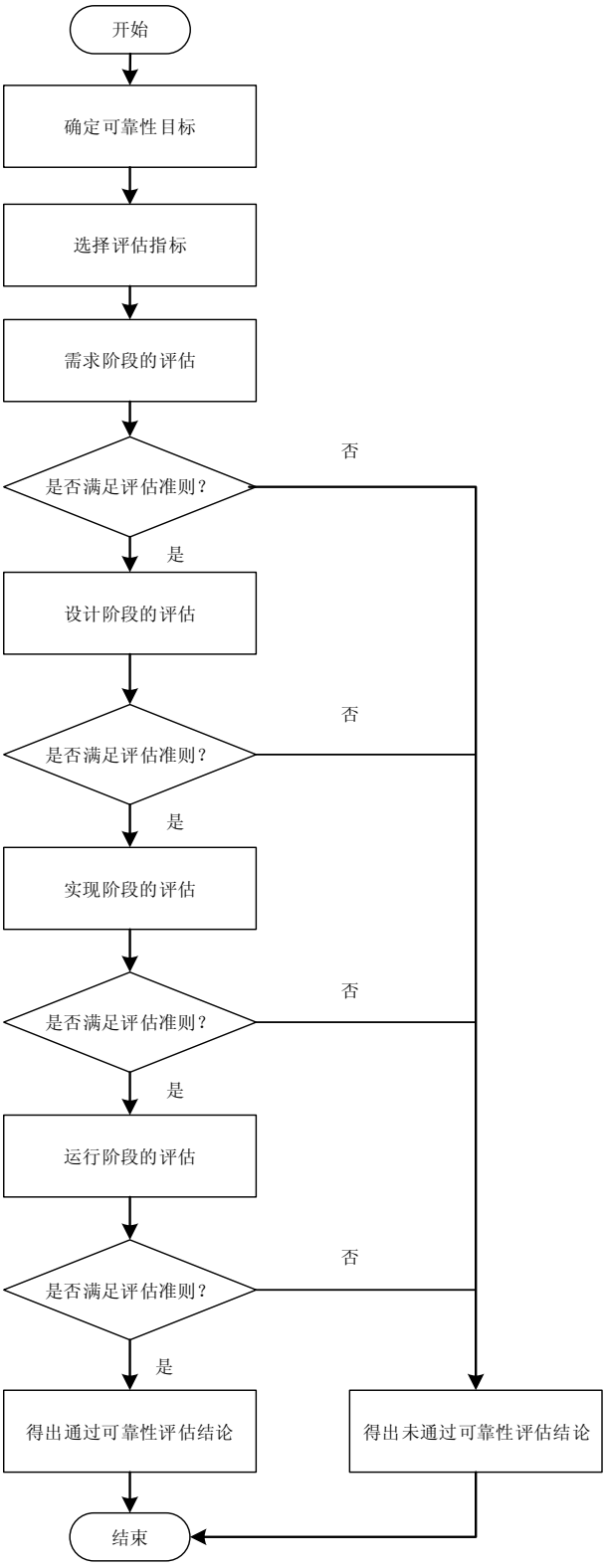

图 2　深度学习算法的可靠性评估流程

4.2　确定可靠性目标

应运用以下步骤确定深度学习算法的可靠性目标：

a) 场景分析

针对深度学习算法实现的功能发生算法失效从而导致软件系统产生一个危险时，需要对其所处的运行环境与运行模式进行描述，既要考虑软件系统正确使用的情况，也要考虑可预见的不正确使用的情况。

b) 危险分析

1) 应通过多种途径开展有关深度学习算法失效的危险识别；如头脑风暴、专家评审会、质量历史记录和软件失效模式和影响分析等技术识别深度学习算法发生算法失效的危害；
2) 应识别危险的后果；如对环境或人员是否有伤害、需要完成的任务是否有影响等；
3) 危险事件应由运行场景和算法失效的相关组合确定；
4) 应以能在深度学习算法所在的软件系统层面观察到的输出来定义结果。

c) 危险严重性等级评估

针对每一个算法失效，应基于确定的理由来预估潜在危险的严重性等级。危险严重性等级（见表1）。

表 1 危险严重性等级

危险严重性等级	描述
灾难级	算法失效导致系统任务失败，或对安全、财产、环境和业务等造成灾难性影响。
严重级	算法失效导致系统任务的主要部分未完成，或对安全、财产、环境和业务等造成严重影响。
一般级	算法失效导致系统完成任务有轻度影响，或对安全、财产、环境和业务等造成一般影响。
轻微级	算法失效导致系统完成任务有障碍但能够完成，或对安全、财产、环境和业务等造成轻微影响或无影响。

危险严重性等级的评估可以基于对多个场景的综合性考虑，同时危险严重性等级的确定应基于场景中有代表性的个体样本。

d) 确定可靠性目标

根据算法失效的危险严重性等级，建立深度学习算法的可靠性目标（见表2）。其中可靠性目标从高到低依次分为A、B、C、D四个级别。

表 2 深度学习算法的可靠性目标

可靠性目标	可靠性目标说明	危险严重性等级对应说明
A	避免算法失效造成灾难级危险	灾难级
B	避免算法失效造成严重级危险	严重级
C	避免算法失效造成一般级危险	一般级
D	避免算法失效造成轻微级危险	轻微级

4.3 选择评估指标

不同可靠性目标的深度学习算法在各个阶段中选取的可靠性评估指标不同，因此在面向算法的需求阶段、设计阶段、实现阶段和运行阶段的可靠性评估过程中应确定与之对应的评估指标。具体选取规则见规范性附录A。

4.4 评估准则

开展可靠性评估工作应遵守以下准则：

a) 各阶段评估通过的准则应同时满足如下要求：
   1) 依据规范性附录A选取的某一级指标下的二级指标全部通过；
   2) 依据规范性附录A选取的某阶段的一级指标全部通过。
b) 深度学习算法可靠性评估通过的准则应满足：面向算法需求阶段、设计阶段、实现阶段及运行阶段四个阶段的可靠性评估均通过。

4.5 各阶段评估

各阶段评估工作应满足：
a) 面向深度学习算法的需求阶段、设计阶段、实现阶段、运行阶段四个阶段实施评估活动；
b) 通过当前阶段的评估是进入下一阶段评估的前提条件之一；
c) 四个阶段的评估活动有完整的顺序关系；
d) 各阶段评估活动的输入、关键活动及输出要求详见本标准第5至第8章；
e) 各阶段可靠性评估结果均应以阶段评估报告的形式进行输出，其内容至少应包括以下内容：
   1) 深度学习算法的可靠性目标；
   2) 开展可靠性评估的阶段名称；
   3) 针对算法在该阶段开展可靠性评估工作所选择的评估指标及针对评估指标的评估结果；
   4) 该阶段的可靠性评估结果。

4.6 评估结论

面向深度学习算法的需求阶段、设计阶段、实现阶段及运行阶段四个阶段均通过评估，深度学习算法可靠性通过评估并达到目标要求；否则未通过评估。

5 需求阶段的评估

5.1 概述

深度学习算法需求阶段是通过调研和分析，理解用户和项目应用的功能、性能等具体要求，最后确定算法应实现的功能性需求、非功能性需求和应满足的设计约束的阶段。

面向深度学习算法需求阶段的可靠性评估工作，指运用可靠性分析方法，通过对算法功能实现的正确性和软硬件平台依赖的影响等进行评估，以确定算法的需求满足可靠性目标要求。

5.2 前提条件

开展本阶段可靠性评估工作前至少应完成获取深度学习算法的可靠性目标。

5.3 输入

开展本阶段可靠性评估工作的输入至少应包括：
a) 软件系统的需求说明书；
b) 系统设计规范；
c) 软硬件接口规范；
d) 深度学习算法的需求；
e) 深度学习算法的功能概念，包括其目标、功能、运行模式及状态；
f) 深度学习算法的运行条件与环境约束。

5.4 关键活动

对应确定后的算法需求阶段的可靠性目标选取评估指标，并从以下关键活动中选取与评估指标对应的关键活动，实施评估工作：

a) 对算法功能实现的正确性进行评估：
   1) 分析需求阶段设定的任务指标要求是否影响可靠性目标；
   2) 分析需求阶段设定的响应时间要求是否影响可靠性目标。
b) 对软硬件平台依赖的影响进行评估：
   1) 分析深度学习框架差异对算法带来的影响；
   2) 分析操作系统差异对算法带来的影响；
   3) 分析硬件架构差异对算法带来的影响。

5.5 输出

深度学习算法需求阶段的可靠性评估报告，评估报告要求见4.5e)。

6 设计阶段的评估

6.1 概述

深度学习算法的设计阶段是根据算法需求阶段得到的需求分析，设计出满足设计约束并能够实现任务功能性需求、非功能性需求的深度学习目标函数及相应的算法，并选取合适的训练数据集的阶段。

面向深度学习算法设计阶段的可靠性评估工作，指运用分析或评审等方法，对算法功能实现的正确性、训练数据集的影响及目标函数等进行评估，以确定算法的设计满足可靠性目标要求。

6.2 前提条件

开展本阶段可靠性评估工作前至少应完成：

a) 深度学习算法需求阶段的可靠性评估工作；
b) 深度学习算法的设计工作。

6.3 输入

开展本阶段可靠性评估工作的输入至少应包括：

a) 深度学习算法需求阶段的可靠性评估报告；
b) 深度学习算法的可靠性评估目标；
c) 深度学习算法的功能说明；
d) 深度学习算法所在的软硬件系统的接口规范；
e) 深度学习算法的训练数据集；
f) 深度学习算法的设计说明。

6.4 关键活动

对应确定后的算法可靠性目标选取评估指标，并从以下关键活动中选取与评估指标对应的关键活动实施评估工作：

a) 对算法功能实现的正确性进行评估：
   1) 分析设计完成后任务指标要求是否满足需求阶段设定的相应要求；
   2) 分析设计完成后响应时间要求是否满足需求阶段设定的相应要求。

b) 对训练数据集进行分析：
   1) 分析训练数据集是否存在不均衡情况；
   2) 分析训练数据集规模是否满足训练需求；
   3) 分析训练数据集标注质量是否满足训练需求；
   4) 分析训练数据集是否受到污染。
c) 对目标函数的影响进行分析：

分析优化目标数量是否满足算法需求。

6.5 输出

深度学习算法设计阶段的可靠性评估报告，评估报告要求见4.5e)。

7 实现阶段的评估

7.1 概述

深度学习算法实现阶段是对算法设计阶段所设计的算法进行编程实现，包括利用数据集对深度学习算法的开展训练、测试与验证等活动。

面向深度学习算法实现阶段的可靠性评估工作，指运用分析和测试等方法，对算法功能实现的正确性、代码实现的正确性、目标函数的影响及对抗性样本的影响等进行评估，以确定算法的实现满足可靠性目标要求。

7.2 前提条件

开展本阶段可靠性评估工作前至少应完成：

a) 深度学习算法设计阶段的可靠性评估工作；
b) 深度学习算法的实现工作。

7.3 输入

开展本阶段可靠性评估工作的输入至少应包括：

a) 深度学习算法需求阶段的可靠性评估报告；
b) 深度学习算法设计阶段的可靠性评估报告；
c) 深度学习算法的可靠性评估目标；
d) 深度学习算法所在的软硬件系统的接口规范；
e) 深度学习算法的训练数据集；
f) 深度学习算法的对抗性样本；
g) 深度学习算法的设计说明；
h) 深度学习算法的功能说明；
i) 深度学习算法的源代码。

7.4 关键活动

对应确定后的算法可靠性目标选取评估指标，并从以下关键活动中选取与评估指标对应的关键活动实施评估工作：

a) 对算法功能实现的正确性进行评估：
   1) 验证算法实现后的任务指标是否达到需求阶段设定的相应要求；

2) 验证算法实现后的响应时间是否达到需求阶段设定的相应要求。

b) 对代码实现的正确性进行评估：
   1) 分析代码是否满足相应的编程规范或指南；
   2) 验证代码是否存在漏洞。

c) 对目标函数的影响进行评估：分析算法的拟合程度对算法可靠性的影响。

d) 对对抗性样本的影响进行分析：
   1) 分析白盒方式生成的样本对算法的影响；
   2) 分析黑盒方式生成的样本对算法的影响；
   3) 分析指定目标方式生成的样本对算法的影响；
   4) 分析不指定目标方式生成的样本对算法的影响。

7.5 输出

深度学习算法实现阶段的可靠性评估报告，评估报告要求见4.5e)。

8 运行阶段的评估

8.1 概述

深度学习算法运行阶段是在实际应用场景下运行包含深度学习算法的软件系统的阶段。

面向深度学习算法运行阶段的可靠性评估工作，指针对实际运行环境使用的数据进行分析，对算法功能实现的正确性、软硬件平台的依赖影响和环境数据的影响等进行评估，以确定算法的运行满足可靠性目标要求。

8.2 前提条件

开展本阶段可靠性评估工作前至少应完成：

a) 深度学习算法实现阶段的可靠性评估工作；
b) 深度学习算法在目标运行环境中的部署工作。

8.3 输入

开展本阶段可靠性评估工作的输入至少应包括：

a) 深度学习算法的可靠性评估目标；
b) 深度学习算法需求阶段的可靠性评估报告；
c) 深度学习算法设计阶段的可靠性评估报告；
d) 深度学习算法实现阶段的可靠性评估报告；
e) 深度学习算法运行中使用的真实数据；
f) 包含深度学习算法的软件系统。

8.4 关键活动

对应确定后的算法可靠性目标选取评估指标，并从以下关键活动中选取与评估指标对应的关键活动实施评估工作：

a) 对算法功能实现的正确性进行评估：
   1) 验证算法运行时任务指标是否达到需求阶段设定的相应要求；
   2) 验证算法运行时响应时间是否达到需求阶段设定的相应要求。

b) 软硬件平台依赖对算法运行的影响：
   1) 分析深度学习框架差异对算法带来的影响；
   2) 分析操作系统差异对算法带来的影响；
   3) 分析硬件架构差异对算法带来的影响。
c) 分析环境数据对算法运行的影响：
   1) 分析环境干扰数据对算法运行的影响，可以参考以下几个方面：
      - 算法输入对象所处环境的复杂情况；
      - 算法输入对象自身环境的复杂情况；
      - 算法输入对象的传输过程的复杂情况；
      - 算法输入对象的数据产品的复杂情况。
   2) 分析数据集分布发生迁移对算法运行的影响；
   3) 分析野值数据对算法运行的影响。

8.5 输出

深度学习算法运行阶段的可靠性报告，评估报告要求见4.5e)。

附 录 A
（规范性附录）
深度学习算法可靠性评估指标选取规则

表A.1给出了深度学习算法的可靠性评估指标的选取规则。针对不同级别的的深度学习算法可靠性目标开展相关评估活动。

表 A.1 选取规则

阶段	可靠性目标	评估指标																			
		算法功能实现的正确性		代码实现的正确性		目标函数的影响		训练数据集的影响				对抗性样本的影响				软硬件平台依赖的影响			环境数据的影响		
		任务指标	响应时间	代码规范性	代码漏洞	优化目标数量	拟合程度	数据集均衡性	数据集规模	数据集标注质量	数据集污染情况	白盒方式生成的样本	黑盒方式生成的样本	指定目标方式生成的样本	不指定目标方式生成的样本	深度学习框架差异	操作系统差异	硬件架构差异	干扰数据	数据集分布迁移	野值数据
需求阶段	A	●	●	--	--	--	--	--	--	--	--	--	--	--	--	●	●	●	--	--	--
	B	●	●	--	--	--	--	--	--	--	--	--	--	--	--	●	●	○	--	--	--
	C	●	○	--	--	--	--	--	--	--	--	--	--	--	--	●	○	○	--	--	--
	D	●	○	--	--	--	--	--	--	--	--	--	--	--	--	○	○	○	--	--	--
设计阶段	A	●	●	--	--	●	--	●	●	●	●	--	--	--	--	--	--	--	--	--	--
	B	●	●	--	--	●	--	●	●	●	○	--	--	--	--	--	--	--	--	--	--
	C	●	○	--	--	○	--	●	●	○	○	--	--	--	--	--	--	--	--	--	--
	D	●	○	--	--	○	--	●	○	○	○	--	--	--	--	--	--	--	--	--	--

表 A.1 （续）

阶段	可靠性目标	评估指标																			
		算法功能实现的正确性		代码实现的正确性		目标函数的影响		训练数据集的影响				对抗性样本的影响				软硬件平台依赖的影响			环境数据的影响		
		任务指标	响应时间	代码规范性	代码漏洞	优化目标数量	拟合程度	数据集均衡性	数据集规模	数据集标注质量	数据集污染情况	白盒方式生成的样本	黑盒方式生成的样本	指定目标方式生成的样本	不指定目标方式生成的样本	深度学习框架差异	操作系统差异	硬件架构差异	干扰数据	数据集分布迁移	野值数据
实现阶段	A	●	●	●	●	--	●	--	--	--	--	●	●	●	●	--	--	--	--	--	--
	B	●	●	●	●	--	●	--	--	--	--	●	●	●	●	--	--	--	--	--	--
	C	●	○	●	○	--	○	--	--	--	--	●	○	○	○	--	--	--	--	--	--
	D	●	○	○	○	--	○	--	--	--	--	○	○	○	○	--	--	--	--	--	--
运行阶段	A	●	●	--	--	--	--	--	--	--	--	--	--	--	--	●	●	●	●	●	●
	B	●	●	--	--	--	--	--	--	--	--	--	--	--	--	●	●	○	●	●	○
	C	●	○	--	--	--	--	--	--	--	--	--	--	--	--	●	○	○	●	○	○
	D	●	○	--	--	--	--	--	--	--	--	--	--	--	--	○	○	○	○	○	○

注： “●”表示对于指定的深度学习算法可靠性目标，必须选择的二级指标；

“○”表示对于指定的深度学习算法可靠性目标，推荐选择的二级指标。

“--”表示不适用。

附 录 B
（资料性附录）
深度学习算法可靠性评估实施案例

表B.1、 B.2分别给出人脸识别算法可靠性评估实施案例和行为检测算法可靠性评估实施案例。

表 B.1 人脸识别算法可靠性评估实施案例

<table>
<tr><td colspan="2">深度学习算法名称</td><td colspan="2">人脸识别</td></tr>
<tr><td colspan="2">深度学习算法说明</td><td colspan="2">人脸识别是基于人的脸部特征信息进行身份识别的一种生物识别技术。其通过摄像机或摄像头采集含有人脸的图像或视频流，自动在图像中检测跟踪人脸，并在人脸数据库中进行人脸检索核实身份，此外还具备活体识别等能力。
人脸识别闸机系统将传统闸机设备集成人脸识别能力，用户在闸机处提供人脸影像进行人脸抽取，采集到的人脸进行云端人脸识别身份验证，闸机随识别验证结果做相应响应。该系统将应用于世博会的门禁系统中。</td></tr>
<tr><td colspan="2">算法可靠性评估方</td><td colspan="2">☑开发方 □用户方 □第三方</td></tr>
<tr><td>第一阶段</td><td colspan="3">确定深度学习算法可靠性目标</td></tr>
<tr><td rowspan="4">1.1 场景分析</td><td>算法运行条件</td><td colspan="2">本地：
硬件设备：智能摄像头，操作系统：Linux，
深度学习框架：PaddlePaddle Mobile，本地设备通过网络接入云端。
云端：
硬件设备:GPU Nvidia P4,操作系统:CentOS 7,深度学习框架:PaddlePaddle Serving</td></tr>
<tr><td>算法运行模式</td><td colspan="2">摄像头捕捉影像，将不同视频帧进行预处理，之后发送云端服务器中部署的人脸识别算法。人脸识别首先判断视频帧中是否包含人脸，如果不存在人脸，切换到下个视频帧。如果存在人脸，则判断是否存在遮挡以及是否为活体人脸。如果存在遮挡等干扰，返回信息提醒除去遮挡物，展示真实人脸。如果人脸正常，算法判断人的身份，如果算法判断的概率在 99%以上，将判断结果发送给闸机，否则提示工作人员协助。如果判断结果为“准入”，打开人脸闸机。否则报警。</td></tr>
<tr><td>正常运行场景</td><td colspan="2">1. 算法持续性接收摄像头前端传送的影像
2. 有人接近摄像头，摄像头捕捉到影像
3. 人脸识别算法判断身份，返回识别结果
4. 用户以人脸作为身份识别凭证进行注册，通过闸机时通过摄像头提供人脸闸机随识别结果做相应响应</td></tr>
<tr><td>可预见的异常场景</td><td colspan="2">1. 人脸存在如太阳镜、口罩等遮挡时的识别
2. 用其他人的图片来请求人脸识别
3. 相似度较高的人脸如双胞胎进行人脸识别</td></tr>
<tr><td rowspan="4">1.2 危险分析</td><td>算法失效序号</td><td>算法失效说明</td><td>识别方法</td></tr>
<tr><td>1</td><td>人脸存在如太阳镜、口罩等遮挡时的识别</td><td>基于类似产品的历史数据</td></tr>
<tr><td>2</td><td>用其他人的图片来请求人脸识别</td><td>头脑风暴</td></tr>
<tr><td>3</td><td>相似度较高的人脸如双胞胎进行人脸识别</td><td>头脑风暴</td></tr>
<tr><td rowspan="4">1.3 危险严重性等级评估</td><td>算法失效序号</td><td>后果</td><td>危险严重性等级</td></tr>
<tr><td>1</td><td>导致重要参会人员无法正常进入会场，需要现场工作人员帮助</td><td>一般</td></tr>
<tr><td>2</td><td>导致不法分子混入会场，可能造成严重后果</td><td>严重</td></tr>
<tr><td>3</td><td>导致不法分子混入会场，可能造成严重后果</td><td>严重</td></tr>
</table>

表 B.1 （续）

1.4 确定可靠性目标	**危险严重性等级说明**			**可靠性目标**
	基于本人脸识别的算法会被应用到国际会议的门禁系统中。会议的参会人员包括各国的政府领导人和商界领袖。算法失效可能导致：1. 重要参会人员无法正常进入会场。2.不法分子混入会议，并造成恶劣的国际影响，所以归为严重级。			B
第二阶段	**选择可靠性评估指标**			
2.1 指标选择说明	**阶段名称**	**选择的二级指标**		
	需求阶段	查准率、查全率、准确率、响应时间、深度学习框架差异、操作系统差异		
	设计阶段	查准率、查全率、准确率、响应时间、优化目标数量、数据集均衡性、数据集规模、数据集标注质量		
	实现阶段	查准率、查全率、准确率、响应时间、代码规范性、代码漏洞、拟合程度、白盒方式生成的样本、黑盒方式生成的样本、指定目标方式生成的样本、不指定目标方式生成的样本		
	运行阶段	查准率、查全率、准确率、响应时间、深度学习框架差异、操作系统差异、干扰数据、数据集分布迁移		
2.2 评估准则说明	a) 各阶段评估通过的准则应同时满足如下要求： 1) 依据规范性选取的某一级指标下的二级指标全部通过。 2) 依据规范性选取的某阶段的一级指标全部通过。 b) 深度学习算法可靠性评估通过的准则应满足：面向算法需求阶段、设计阶段、实现阶段及运行阶段四个阶段的可靠性评估均通过。			
第三阶段	**面向算法需求阶段的可靠性评估**			
3.1 输入说明	软件系统的需求说明书：人脸识别系统需求说明书-V1.0			
	软件系统设计规范：人脸识别系统设计规范-V1.0			
	软硬件接口规范：人脸闸机系统软硬件接口规范-V1.0			
	深度学习算法的需求说明书：人脸检测识别算法需求说明书-V1.0			
3.2 关键活动	**一级指标**	**算法功能实现的正确性**		
	二级指标	**名称**	**评估工作**	**评估结果**
		查准率	查准率阈值预计为 99.9%以上，经专家评审和技术负责人确认满足需求；类似软件系统中的要求为 99%以上，达到相应要求。	☑通过☐未通过
		查全率	查全率阈值预计为 99.9%以上，经专家评审和技术负责人确认满足需求；类似软件系统中的要求为 99%以上，达到相应要求。	☑通过☐未通过
		准确率	准确率阈值预计为 99.9%以上，经专家评审和技术负责人确认满足需求；类似软件系统中的要求为 99%以上，达到相应要求。	☑通过☐未通过
		响应时间	响应时间阈值预计为 20ms 以内，经专家评审和技术负责人确认满足需求；类似软件系统中的要求为 50ms 以内，达到相应要求。	☑通过☐未通过
	一级指标评估结果	☑通过 ☐未通过		
	一级指标	**软硬件平台依赖的影响**		
	二级指标	**名称**	**评估工作**	**评估结果**
		深度学习框架差异	所选深度学习框架 PaddlePaddle 支持 CPU、GPU 多种设备及其混布计算，分布式训练和预测性能良好，提供 C++和 Python 两种高层 API 易于使用，可以方便的进行 Linux, IOS 和 Android 移动端环境上的编译部署，满足适用性要求。	☑通过☐未通过
		操作系统差异	移动端所使用操作系统为 Linux, IOS 或 Android，云端所使用的操作系统类型为 Linux，可以运行在 CentOS 7 或者 Ubuntu 16.04 上，系统安全稳定。对于其他版本有潜在移植风险的操作系统，亦可使用 Docker 进行服务部署，不存在因可移植导致失效的风险。	☑通过☐未通过
	一级指标评估结果	☑通过 ☐未通过		

表 B.1 （续）

3.3 输出	**面向算法需求阶段的可靠性评估** ☑通过 □未通过			
第四阶段	**面向算法设计阶段的可靠性评估**			
4.1 输入说明	深度学习算法的训练数据集：全球人脸数据集-A			
	深度学习算法的设计说明：人脸识别算法设计文档-V1.0			
4.2 关键活动	**一级指标**	**算法功能实现的正确性**		
	二级指标	**名称**	**评估工作**	**评估结果**
		查准率	设计后的查准率阈值预计为 99.9%，经过评审，能够达到需求阶段设计的要求。	☑通过□未通过
		查全率	设计后的查全率阈值预计为 99.9%，经过评审，能够达到需求阶段设计的要求。	☑通过□未通过
		准确率	设计后的准确率阈值预计为 99.9%，经过评审，能够达到需求阶段设计的要求。	☑通过□未通过
		响应时间	设计后的响应时间阈值预计为 12ms，经过评审，能够达到需求阶段设计的要求。	☑通过□未通过
	一级指标评估	☑通过 □未通过		
	一级指标	**训练数据集的影响**		
	二级指标	**名称**	**评估工作**	**评估结果**
		数据集均衡性	数据集为“全球人脸数据集-A”，数据包含 1.100 个主要人口国家的人脸数据； 2.70%正脸，30%侧脸； 3.50%无遮挡，30%轻微遮挡，20%严重遮挡； 4.50%男性，50%女性； 5.0~18 岁 20%，18~40 岁 30%，40~60 岁 30%，60 岁以上 20%； 6.正常人脸 80%，带有伤疤能不可除去遮挡 20%。	☑通过□未通过
		数据集规模	数据集规模约为 100 万张。其中利用 50 万张数据样本可以达到预期可靠性，并且无明显过拟合和欠拟合。使用超过 100 万数据后，训练时间变长但算法效果无明显提升。	☑通过□未通过
		数据集标注质量	数据集中 100%的数据完成标注，并且 99.99%的样本标注准确	☑通过□未通过
	一级指标评估	☑通过 □未通过		
	一级指标	**目标函数的影响**		
	二级指标	**名称**	**评估工作**	**评估结果**
		优化目标数量	通过与目前主流的类似算法的对比，现阶段优化目标数量不存在过少或过多的情况	☑通过□未通过
	一级指标评估	☑通过 □未通过		
4.3 输出	**面向算法设计阶段的可靠性评估** ☑通过 □未通过			

表 B.1 （续）

<table>
<tr><td>第五阶段</td><td colspan="4">面向算法实现阶段的可靠性评估</td></tr>
<tr><td>5.1 输入说明</td><td colspan="4">深度学习算法的对抗性样本数据集：人脸对抗样本集-A</td></tr>
<tr><td></td><td colspan="4">深度学习算法的源代码：人脸识别算法源码-V1.0</td></tr>
<tr><td rowspan="15">5.2 关键活动</td><td>一级指标</td><td colspan="3">算法功能实现的正确性</td></tr>
<tr><td rowspan="5">二级指标</td><td>名称</td><td>评估工作</td><td>评估结果</td></tr>
<tr><td>查准率</td><td>实现后的查准率 99.98%，经过评审，能够达到设计阶段设定的要求。</td><td>☑通过☐未通过</td></tr>
<tr><td>查全率</td><td>实现后的查全率为 99.97%，经过评审，能够达到设计阶段设定的要求。</td><td>☑通过☐未通过</td></tr>
<tr><td>准确率</td><td>实现后的准确率为 99.96%，经过评审，能够达到设计阶段设定的要求。</td><td>☑通过☐未通过</td></tr>
<tr><td>响应时间</td><td>实现后的响应时间为 11ms，经过评审，能够达到设计阶段设定的要求。</td><td>☑通过☐未通过</td></tr>
<tr><td>一级指标评估</td><td colspan="3">☑通过　☐未通过</td></tr>
<tr><td>一级指标</td><td colspan="3">代码实现的正确性</td></tr>
<tr><td rowspan="3">二级指标</td><td>名称</td><td>评估工作</td><td>评估结果</td></tr>
<tr><td>代码规范性</td><td>代码编写遵循 Google C++/Python Code-style 代码规范，并使用 cpp-lint/clang-format/yarf 等工具进行了代码规范性的检查和补充完善。</td><td>☑通过☐未通过</td></tr>
<tr><td>代码漏洞</td><td>代码经由 Pprof, valgrind 和 QA 测试，测试覆盖率 90%+。</td><td>☑通过☐未通过</td></tr>
<tr><td>一级指标评估</td><td colspan="3">☑通过　☐未通过</td></tr>
<tr><td>一级指标</td><td colspan="3">目标函数的影响</td></tr>
<tr><td rowspan="2">二级指标</td><td>名称</td><td>评估工作</td><td>评估结果</td></tr>
<tr><td>拟合程度</td><td>训练过程，未出现显著过拟合或者欠拟合情况</td><td>☑通过☐未通过</td></tr>
<tr><td>一级指标评估</td><td colspan="3">☑通过　☐未通过</td></tr>
<tr><td>一级指标</td><td colspan="3">对抗性样本的影响</td></tr>
<tr><td rowspan="5">二级目标</td><td>名称</td><td>评估工作</td><td>评估结果</td></tr>
<tr><td>白盒方式生成的样本</td><td>生成样本数据 2 万个，抗噪性能提升 90%。</td><td>☑通过☐未通过</td></tr>
<tr><td>黑盒方式生成的样本</td><td>生成样本数据 2 万个，抗噪性能提升 80%。</td><td>☑通过☐未通过</td></tr>
<tr><td>指定目标方式生成的样本</td><td>生成样本数据 2 万个，抗噪性能提升 90%。</td><td>☑通过☐未通过</td></tr>
<tr><td>不指定目标方式生成的样本</td><td>生成样本数据 2 万个，抗噪性能提升 80%。</td><td>☑通过☐未通过</td></tr>
<tr><td>一级指标评估</td><td colspan="3">☑通过　☐未通过</td></tr>
<tr><td>5.3 输出</td><td colspan="4">面向算法实现阶段的可靠性评估　☑通过　☐未通过</td></tr>
<tr><td>第六阶段</td><td colspan="4">面向算法运行阶段的可靠性评估</td></tr>
<tr><td>6.1 输入说明</td><td colspan="4">深度学习算法运行中使用的数据：实时采集摄像头捕捉的影像</td></tr>
<tr><td></td><td colspan="4">包含深度学习算法的软件系统：人脸识别系统-V1.0</td></tr>
</table>

表 B.1 （续）

6.2 关键活动	一级指标	算法功能实现的正确性		
	二级指标	名称	评估工作	评估结果
		查准率	算法实际运行 30 天时间中查准率为 99.98%，经过评审，能够达到设计阶段设定的要求。	☑通过□未通过
		查全率	算法实际运行 30 天时间中查全率为 99.97%，经过评审，能够达到设计阶段设定的要求。	☑通过□未通过
		准确率	算法实际运行 30 天时间中准确率为 99.96%，经过评审，能够达到设计阶段设定的要求。	☑通过□未通过
		响应时间	算法实际运行 30 天时间中平均单次识别响应时间为 11ms，经过评审，能够达到设计阶段设定的要求。	☑通过□未通过
	一级指标评估	☑通过 □未通过		
	一级指标	软硬件平台依赖的影响		
	二级指标	名称	评估工作	评估结果
		深度学习框架差异	所选深度学习框架 PaddlePaddle 提供的 C++接口和执行引擎易于编译集成到实际运行环境中，识别延迟在 20ms 内，满足设定的目标。	☑通过□未通过
		操作系统差异	实际运行阶段，算法运行在 CentOS7 中，系统可以稳定长时间运行，并无安全漏洞，不潜在失效的风险。	☑通过□未通过
	一级指标评估	☑通过 □未通过		
	一级指标	环境数据影响		
	二级指标	名称	评估工作	评估结果
		干扰数据	算法实际运行 30 天时间中，测试了超过 1 万张实际人脸，未发生干扰数据影响算法正确运行的情况。	☑通过□未通过
		数据集分布迁移	算法实际运行 30 天时间中，实际输入数据集分布和训练测试数据集分布较为近似，系统未发生失效。	☑通过□未通过
	一级指标评估	☑通过 □未通过		
6.3 输出	面向算法运行阶段的可靠性评估 ☑通过 □未通过			
第七阶段	深度学习算法的可靠性评估			
评估结论	面向深度学习算法的需求阶段、设计阶段、实现阶段及运行阶段四个阶段均通过评估，深度学习算法可靠性通过评估并达到 B 级目标要求。			

表 B.2　行为检测算法可靠性评估实施案例

<table>
<tr><td>深度学习算法名称</td><td colspan="3">行为检测</td></tr>
<tr><td>深度学习算法说明</td><td colspan="3">行为检测技术是对监控视频中装载货物的工作人员进行行为判断，是否出现对包裹进行扔、抛等不友好的暴力行为。具体实现是通过对输入的视频单帧图像，首先检测图像中的人是否手拿包裹，以开始出现拿包裹的人图像作为起始帧，再对拿包裹的人进行轨迹追踪与行为分类，从而进行行为识别。
行为检测系统是将行为检测技术集成至一个小型高性能计算盒子中，场院的监控视频通过交换机传输到盒子后，搭载在盒子上的行为检测技术对监控视频进行行为识别，识别出来的结果传输到云端进行事件的管控（如预警）。现该系统已经运用在物流企业多个作业区域。</td></tr>
<tr><td>算法可靠性评估方</td><td colspan="3">☑开发方　　□用户方　　□第三方</td></tr>
<tr><td>第一阶段</td><td colspan="3">确定深度学习算法可靠性目标</td></tr>
<tr><td rowspan="4">1.1 场景分析</td><td>算法运行条件</td><td colspan="2">本地：
硬件设备：普通摄像头、高性能计算盒子（NVIDIA TX2 1.5TFlops）
操作系统：Linux 3.10.0, 深度学习框架：Tensorflow 1.4.0
云端：
硬件设备：48Core intel,256G 内存，2T 浪潮服务器集群
操作系统：CentOS, 深度学习框架：Tensorflow 1.4.0</td></tr>
<tr><td>算法运行模式</td><td colspan="2">摄像头捕捉到监控视频后，通过交换机传输到盒子后，搭载在盒子上的行为检测技术对监控视频进行实时行为检测，检测出来的结果传输到云端进行事件的管控（如预警）。行为检测整体算法流程可分为数据准备、检测和行为分类三个阶段。
数据准备包含了数据收集、数据标注和数据预处理三个步骤。数据收集即对原始监控视频的采集；数据标注则是标出监控视频图像中所有接触货物和不接触货物的两类行人的坐标；数据预处理包含数据归一化以及数据增强。
检测是对于监控视频中的连续帧，按顺序对每一帧进行行人检测，以存在接触包裹的行人的帧为起始帧，接触包裹的行人的坐标为空间位置，从起始帧开始往后取固定帧数（如 15 帧），就可以提取出了暴力分拣行为识别所需要的候选区域。
行为分类是对侯选区域进行行为分类（如是否暴力行为），如果分类结果为暴力，则该事件将上传至云端，触发事件预警；否则重新回到下一起始帧计算。</td></tr>
<tr><td>正常运行场景</td><td colspan="2">1. 监控摄像头成像正常
2. 视野内有待检测体
3. 行为检测系统正常检测出候选区域并上传检测结果至云端</td></tr>
<tr><td>可预见的异常场景</td><td colspan="2">1. 摄像头成像异常，如帧率过低
2. 摄像头视野内无检测体
3. 图像传输过程丢失信息</td></tr>
<tr><td rowspan="4">1.2 危险分析</td><td>算法失效序号</td><td>算法失效说明</td><td>识别方法</td></tr>
<tr><td>1</td><td>摄像头成像异常，如帧率过低</td><td>专家评审</td></tr>
<tr><td>2</td><td>图像传输过程丢失信息</td><td>质量历史记录</td></tr>
<tr><td>3</td><td>摄像头视野内无检测体</td><td>质量历史记录</td></tr>
<tr><td rowspan="4">1.3 危险严重性等级评估</td><td>算法失效序号</td><td>后果</td><td>危险严重性等级</td></tr>
<tr><td>1</td><td>导致检测行为准确率偏低，存在漏检隐患</td><td>一般级</td></tr>
<tr><td>2</td><td>导致检测行为准确率偏低，存在漏检隐患</td><td>一般级</td></tr>
<tr><td>3</td><td>导致无法检测行为，存在操作安全隐患</td><td>严重级</td></tr>
</table>

表 B.2 （续）

<table>
<tr><td rowspan="2">1.4 确定可靠性目标</td><td colspan="3">危险严重性等级说明</td><td>可靠性目标</td></tr>
<tr><td colspan="3">基于行为检测技术的行为检测系统将会被运用到多种环境下作业区域的多种行为检测（暴力、着装、礼仪等），多种环境包含快递、银行等作业区域。算法失效可能导致：1）检测行为准确率偏低，存在行为漏检隐患；2）无法检测行为，存在操作安全隐患。</td><td>B</td></tr>
<tr><td>第二阶段</td><td colspan="4">选择可靠性评估指标</td></tr>
<tr><td rowspan="5">2.1 指标选择说明</td><td>阶段名称</td><td colspan="3">选择的二级指标</td></tr>
<tr><td>需求阶段</td><td colspan="3">准确率、召回率、响应时间、深度学习框架差异、操作系统差异</td></tr>
<tr><td>设计阶段</td><td colspan="3">准确率、召回率、响应时间、优化目标数量、数据集均衡性、数据集规模、数据集标注质量</td></tr>
<tr><td>实现阶段</td><td colspan="3">准确率、召回率、响应时间、代码规范性、代码漏洞、拟合程度、白盒方式生成的样本、黑盒方式生成的样本、指定目标方式生成的样本、不指定目标方式生成的样本</td></tr>
<tr><td>运行阶段</td><td colspan="3">准确率、召回率、响应时间、深度学习框架差异、操作系统差异、干扰数据、数据集分布迁移</td></tr>
<tr><td>2.2 评估准则说明</td><td colspan="4">a） 各阶段评估通过的准则应同时满足如下要求：
1） 依据规范性选取的某一级指标下的二级指标全部通过；
2） 依据规范性选取的某阶段的一级指标全部通过。
b） 深度学习算法可靠性评估通过的准则应满足：面向算法需求阶段、设计阶段、实现阶段及运行阶段四个阶段的可靠性评估均通过。</td></tr>
<tr><td>第三阶段</td><td colspan="4">面向算法需求阶段的可靠性评估</td></tr>
<tr><td rowspan="4">3.1 输入说明</td><td colspan="4">软件系统的需求说明书：行为检测系统需求文档_v1.0.docx</td></tr>
<tr><td colspan="4">软件系统设计规范：行为检测系统设计规范_v1.0.docx</td></tr>
<tr><td colspan="4">软硬件接口规范：行为检测系统接口规范_v1.0.docx</td></tr>
<tr><td colspan="4">深度学习算法的需求说明书：行为检测算法需求说明_v1.0.docx</td></tr>
<tr><td rowspan="10">3.2 关键活动</td><td>一级指标</td><td colspan="3">算法功能实现的正确性</td></tr>
<tr><td rowspan="4">二级指标</td><td>名称</td><td>评估工作</td><td>评估结果</td></tr>
<tr><td>准确率</td><td>准确率阈值预计为 0.95,经技术负责人 确认满足需求；类似软件系统中的要求为 0.90，达到相应要求。</td><td>☑通过☐未通过</td></tr>
<tr><td>召回率</td><td>召回率阈值预计为 0.95，经技术负责人确认满足需求；类似软件系统中的要求为 0.90，达到相应要求。</td><td>☑通过☐未通过</td></tr>
<tr><td>响应时间</td><td>响应时间阈值预计为 0.3s，经技术负责人确认满足需求;类似软件系统中的要求为 0.5s，达到相应要求。</td><td>☑通过☐未通过</td></tr>
<tr><td>一级指标评估结果</td><td colspan="3">☑通过 ☐未通过</td></tr>
<tr><td>一级指标</td><td colspan="3">软硬件平台依赖的影响</td></tr>
<tr><td rowspan="3">二级指标</td><td>名称</td><td>评估工作</td><td>评估结果</td></tr>
<tr><td>深度学习框架差异</td><td>基于 caffe 和 tensorflow 框架，集成于嵌入式平台，支持高清视频流的实时识别。TensorFlow 框架满足在 Python 、接口设计、分布式性能等的适用性要求。</td><td>☑通过☐未通过</td></tr>
<tr><td>操作系统差异</td><td>本地：系统运行嵌入式平台上的 ubuntu 稳定版系统，漏洞少，可稳定长时间执行，且易于 3D 打印，迅速部署。
云端：运行于 centos 服务器版。</td><td>☑通过☐未通过</td></tr>
<tr><td>一级指标评估结果</td><td colspan="3">☑通过 ☐未通过</td></tr>
</table>

表 B.2 （续）

3.3 输出	**面向算法需求阶段的可靠性评估** ☑通过 ☐未通过			
第四阶段	**面向算法设计阶段的可靠性评估**			
4.1 输入说明	深度学习算法的训练数据集：作业区域行人操作视频训练集_v1.0.zip			
	深度学习算法的设计说明：行为检测系统算法设计说明_v1.0.docx			
4.2 关键活动	**一级指标**	**算法功能实现的正确性**		
	二级指标	**名称**	**评估工作**	**评估结果**
		准确率	设计后的准确率阈值预计为 0.95,经过评审，能够达到需求阶段设计的要求。	☑通过☐未通过
		召回率	设计后的召回率阈值预计为 0.95,经过评审，能够达到需求阶段设计的要求。	☑通过☐未通过
		响应时间	设计后的响应时间阈值预计为 0.3s，经过评审，能够达到需求阶段设计的要求。	☑通过☐未通过
	一级指标评估	☑通过 ☐未通过		
	一级指标	**训练数据集的影响**		
	二级指标	**名称**	**评估工作**	**评估结果**
		数据集均衡性	数据集为作业区域行人操作视频，其中包括 1. 50 个点部 2 天装货区域正常作业视频共计 10 万个； 2. 50 个点部 2 天装货区域非正常作业视频共计 10 万个； 3. 50 个点部 2 天卸货区域政策作业视频共计 10 万个； 4. 50 个点部 2 天卸货区域非正常作业视频共计 10 万个； 5. 员工办公场所 2 天作业视频共计 5 万个（含正负样本）。	☑通过☐未通过
		数据集规模	视频流数据集规模为一百万视频集合。约五十万视频数据作为训练样本即可达到预期的可靠性。超过一百万训练视频后准确率无明显提升。	☑通过☐未通过
		数据集标注质量	数据集中 100%的数据完成标注，并且 98%的样本标注准确。	☑通过☐未通过
	一级指标评估	☑通过 ☐未通过		
	一级指标	**目标函数的影响**		
	二级指标	**名称**	**评估工作**	**评估结果**
		优化目标数量	对优化目标数量进行评估，通过评审会/与类似算法对比，现阶段优化目标数量不存在过少或过多的情况。	☑通过☐未通过
	一级指标评估	☑通过 ☐未通过		
4.3 输出	**面向算法设计阶段的可靠性评估** ☑通过 ☐未通过			

表 B.2 （续）

第五阶段	面向算法实现阶段的可靠性评估			
5.1 输入说明	深度学习算法的对抗性样本：行为检测系统视频训练集_v1.0.zip			
	深度学习算法的设计说明：行为检测系统算法设计说明_v1.0.docx			
	深度学习算法的功能说明：行为检测系统算法功能说明_v1.0.docx			
	深度学习算法的源代码：行为检测系统算法源代码_v1.0.zip			
5.2 关键活动	一级指标	算法功能实现的正确性		
	二级指标	名称	评估工作	评估结果
		准确率	实现后的准确率为 0.96，经过评审，能够达到设计阶段设定的要求	☑通过☐未通过
		召回率	实现后的召回率为 0.96，经过评审，能够达到设计阶段设定的要求	☑通过☐未通过
		响应时间	实现后的响应时间为 0.3s，经过评审，能够达到设计阶段设定的要求	☑通过☐未通过
	一级指标评估	☑通过 ☐未通过		
	一级指标	代码实现的正确性		
	二级指标	名称	评估工作	评估结果
		代码规范性	代码依据 PEP8 规范编写，且遵循 python 代码礼仪，git flow 等规范。并使用 pylint、flake8 进行代码检查。	☑通过☐未通过
		代码漏洞	经过 Sonar 代码扫描、模型压力测试、AB 测试等，代码走查覆盖率超 90%。	☑通过☐未通过
	一级指标评估	☑通过 ☐未通过		
	一级指标	目标函数的影响		
	二级指标	名称	评估工作	评估结果
		拟合程度	模型准确率等各个指标在训练集以及测试集中数值相近，不存在的过拟合的情况，同时各个指标达到预期，不存在欠拟合的情况	☑通过☐未通过
	一级指标评估	☑通过 ☐未通过		
	一级指标	对抗性样本的影响		
	二级目标	名称	评估工作	评估结果
		白盒方式生成的样本	生成样本数据 10000 个，数据集名称白盒测试集，算法训练后针对视频质量不佳情况提升了 0.1 准确率指标。	☑通过☐未通过
		黑盒方式生成的样本	生成样本数据 10000 个，数据集名称黑盒测试集，算法训练后针对视频质量不佳情况提升了 0.05 准确率指标。	☑通过☐未通过
		指定目标方式生成的样本	生成样本数据 3000 个，数据集名称指定目标测试集，算法训练后针对视频质量不佳情况提升了 0.01 准确率指标。	☑通过☐未通过
		不指定目标方式生成的样本	生成样本数据 3000 个，数据集名称不指定目标测试集，算法训练后针对视频质量不佳情况提升了 0.01 准确率指标。	☑通过☐未通过
	一级指标评估	☑通过 ☐未通过		
5.3 输出	面向算法实现阶段的可靠性评估 ☑通过 ☐未通过			
第六阶段	面向算法运行阶段的可靠性评估			

表 B.2 （续）

6.1 输入说明	深度学习算法运行中使用的数据：行为检测视频测试数据_v1.0.zip			
	包含深度学习算法的软件系统：行为检测识别系统_v1.0			
6.2 关键活动	**一级指标**	**算法功能实现的正确性**		
	二级指标	**名称**	**评估工作**	**评估结果**
		准确率	该算法所在的系统实际运行 30 天后，根据实际运行的数据统计为 0.97。	☑通过☐未通过
		召回率	该算法所在的系统实际运行 30 天后，根据实际运行的数据统计为 0.97。	☑通过☐未通过
		响应时间	该算法所在的系统实际运行 30 天后，根据实际运行的数据统计为 0.3s。	☑通过☐未通过
	一级指标评估	☑通过　☐未通过		
	一级指标	**软硬件平台依赖的影响**		
	二级指标	**名称**	**评估工作**	**评估结果**
		深度学习框架差异	该算法所在的系统实际运行 30 天后，其选取的深度学习框架、接口设计、分布式性能等的满足达到指标，计算速度达到实时性。	☑通过☐未通过
		操作系统差异	实际运行 30 天中，涉及到的操作系统包括 Linux 3.10.0 等，未发现可操作性、可移植性、系统安全等问题，计算速度达到实时性。	☑通过☐未通过
	一级指标评估	☑通过　☐未通过		
	一级指标	**环境数据影响**		
	二级指标	**名称**	**评估工作**	**评估结果**
		干扰数据	算法实际运行 30 天中，遇到各种情况（环境复杂、自身复杂、传输过程异常、产品异常等）的数据，系统的并未发生失效。	☑通过☐未通过
		数据集分布迁移	算法实际运行 30 天中,实际输入数据集分布，系统的并未发生失效。	☑通过☐未通过
	一级指标评估	☑通过　☐未通过		
6.3 输出	**面向算法运行阶段的可靠性评估**　☑通过　☐未通过			
第七阶段	**深度学习算法的可靠性评估**			
评估结论	面向深度学习算法的需求阶段、设计阶段、实现阶段及运行阶段四个阶段均通过评估，深度学习算法可靠性通过评估并达到 B 级目标要求			

参 考 文 献

[1] GJB/Z 161-2012 军用软件可靠性评估指南
[2] GB/T 16260.2-2006 软件工程 产品质量 第2部分：外部质量
[3] DO-178C-2011 Software Consideration in Airborne Systems and Equipment Certification
[4] GB/T 5271.28-2001 信息技术 词汇 第28部分:人工智能 基本概念与专家系统
[5] 周志华 机器学习 清华大学出版社 2016