

Задачи для Летней интернатуры Intel 2018 в Нижнем Новгороде

Contents

- Задача 1. Оптимизация производительности C++/Fortran-приложения с помощью Intel® Advisor
- Задача 2. Улучшение Python API в Intel® Advisor
- Задача 3. Разработка рекомендации по оптимизации векторизованного кода с использованием Intel® Advisor
- Задача 4. Разработка инструментов профилирования загрузки операционной системы для беспилотного автомобиля
- Задача 5. Логика интеграции коллективных операций для следующего поколения продукта Intel®MPI
- Задача 6. Разработка и оптимизация функций изменения размера изображения (Resize) на основе арифметики с фиксированной точкой
- Задача 7. Разработка стабильного, точного и простого в использовании инструмента для измерения производительности математических функций
- Задача 8. Продуктизация приложения для визуализации результатов обработки изображений с помощью функций из библиотеки Intel® IPP
- Задача 9. Интеграция Intel® DAAL в TensorFlow*
- Задача 10. Разработка процесса поставок программных продуктов по запросам заказчиков вне расписания в соответствии с инженерными и продуктовыми требованиями
- Задача 11. Расширение функциональности компоненты Flow Graph библиотеки Intel® Threading Building Blocks (Intel® TBB)
- Задача 12. Реализация поддержки возможностей последних версий стандарта C++ в Intel® TBB
- Задача 13. Интеграционное тестирование алгоритмов классического машинного обучения в Intel® DAAL
- Задача 14. Алгоритмическая оптимизация нейронных сетей
- Задача 15. Исследование и разработка библиотеки глубокого обучения при решении задачи семантической сегментации
- Задача 16. Реализация и сравнение стратегий распределения вычислений и данных на аппаратной платформе Intel® Movidius™ в задачах вычисления моделей глубокого обучения
- Задача 17. Портирование алгоритма компьютерного зрения на ПЛИС
- Задача 18. Внедрение алгоритмов компьютерного зрения в приложение
- Задача 19. Разработка структур данных для систем с гетерогенной памятью
- Задача 20. Модифицировать структуры данных библиотеки Intel® TBB для использования энергонезависимой памяти
- Задача 21. Микроархитектурная оптимизация приложений для высокопроизводительных систем
- Задача 22. Визуализатор для энкодеров Intel®Media SDK

- Задача 23. Разработка модуля для визуализации 3D-модели окружающей среды автономного автомобиля с использованием Web-технологий
- Задача 24. Анализ производительности параллельных алгоритмов стандартной библиотеки C++ (Parallel STL)
- Задача 25. Осуществление поддержки нейронных сетей в формате хранения ONNX в OpenCV.
- Задача 26. Исследование и разработка алгоритмов глубокого обучения для задачи детектирования объектов
- Задача 27. Оптимизация OpenCV для новых способов организации вычислений
- Задача 28. Интеграция кодека AV1 в кроссплатформенный свободный продукт для конвертации форматов видеофайлов HandBrake
- Задача 29. Анализ изображений с целью извлечения мультиспектральных данных
- Задача 30. Разработка систем для автоматизации тренировки и тестирования алгоритмов глубокого обучения на кластере
- Задача 31. Интеграция инструмента контроля корректности видеокодирования ASG (Artificial Stream Generator) в open-source экосистему Intel® Media Server Studio for Linux*
- Задача 32. Стенд для демонстрации возможностей процессоров Intel в приложениях реального времени

Задача 1. Оптимизация производительности C++/Fortran-приложения с помощью Intel® Advisor

Описание. Intel® Advisor - это инструмент, позволяющий собирать множество данных о производительности приложения и потенциальных оптимизациях. Требуется оптимизировать существующий код на C++, эффективно используя средства Intel – компилятор и Advisor. Необходимо изучить различные компиляторные оптимизации, векторизацию, инструмент Intel® Advisor и его последние возможности для подготовки обучающего материала. Кроме того, необходимо протестировать производительность приложения на различных конфигурациях, используя облачные решения Amazon* и Google*. В качестве дополнительного задания может рассматриваться написание аналогичного кода на языке Fortran для дальнейшего сравнения производительности.

Требования к претенденту. Знание C/C++, Visual Studio, параллельного программирования, векторизации.

Задача 2. Улучшение Python API в Intel® Advisor

Описание. Intel® Advisor - это инструмент, позволяющий собирать множество данных о производительности приложения и потенциальных оптимизациях. Помимо обычных GUI и CLI, Advisor также предоставляет Python API для доступа к данным для тех пользователей, кто хочет самостоятельно анализировать данные. Требуется улучшить Python API так, чтобы им было проще пользоваться и легче использовать вместе с современными библиотеками по анализу данных и машинному обучению в Python.

Требования к претенденту. Знание Python, C/C++.

Опыт применения Python в проектах машинного обучения или анализа данных является дополнительным плюсом.

Задача 3. Разработка рекомендации по оптимизации векторизованного кода с использованием Intel® Advisor

Описание. Intel® Advisor – профилировщик для анализа векторизованных циклов и оптимизации приложений, входящий в состав Intel® Parallel Studio. Необходимо исследовать

проблему производительности, связанную с подготовкой исполнения векторизованного кода, разработать алгоритмы её детектирования и интегрировать их в Intel® Advisor.

Требования к претенденту. Понимание различных техник оптимизации кода.

Желательно хорошее знание C++.

Знание x86 архитектуры и ассемблера будет дополнительным преимуществом кандидата.

Задача 4. Разработка инструментов профилирования загрузки операционной системы для беспилотного автомобиля

Описание. В системах содействия водителю (система автопилотирования автомобиля) важным является соблюдение требований надежности и безопасности программного обеспечения. Для операционной системы реального времени это означает в том числе и то, что она должна гарантировать выполнение своих функций за определенное время. Не менее важным является быстрая загрузка самой системы – чем быстрее она начнет выполнять свои функции, тем быстрее автомобиль сможет начать движение.

Современные процессоры Intel® предоставляют доступ к внутрисхемной отладке (JTAG) с помощью технологии Intel Direct Connect Interface (DCI). Она позволяет подключиться к компьютеру с помощью USB3.0 кабеля и дает полный доступ к процессору, оперативной памяти, контроллерам и т.д.

С помощью данной технологии можно считывать различные метрики (например загруженность ядер процессора), получать данные об исполняемых процессах, а также записывать весь исполненный на процессоре код с точностью до условного и косвенного перехода с помощью технологии Intel® Processor Trace (Intel® PT).

Собрав такую информацию во время запуска операционной системы, можно восстановить ход событий, определить степень использования ресурсов и найти потенциальные проблемы производительности.

Необходимо реализовать приложение, которое через отладочный интерфейс будет периодически собирать данные об использовании процессора и отображать эту информацию на графиках.

Требования к претенденту. Знание Python, C/C++, интерес к работе с процессорами на самом нижнем уровне.

Задача 5. Логика интеграции коллективных операций для следующего поколения продукта Intel®MPI

Описание. Библиотека MPI реализует коммуникации точка-точка и коллективные операции (один ко многим, многие к одному, иногда ко многим), которые должны работать максимально быстро на современных сетевых интерфейсах Intel (например, Omni-Path). Скорость коммуникаций сильно зависит от алгоритмов, реализованных для определенных топологий и размеров сообщений.

Основной частью задачи является разработка и реализация инновационной логики регистрации алгоритмов коллективных операций, которая позволила бы свести к минимуму влияние на код во время интеграции. Другой частью задачи является реализация новых алгоритмов с целью их дальнейшей интеграции в продукт.

Требования к претенденту. Знание C, Perl, Python, регулярные выражения. Желательны базовые знания Autotools и MPI стандарта.

Задача 6. Разработка и оптимизация функций изменения размера изображения (Resize) на основе арифметики с фиксированной точкой

Описание. Изменение размера изображений – одна из наиболее часто встречаемых операций в алгоритмах обработки изображений и компьютерного зрения. Некоторые реализации направлены на получение выходного изображения с максимальной точностью, другие же наоборот жертвуют качеством ради увеличения скорости исполнения.

Требуется разработать функцию изменения размера изображений со сглаживанием на основе арифметики с фиксированной точкой и их оптимизации для процессоров Intel.

Решение задачи дает реальный шанс разработать самый быстрый Resize со сглаживанием, который впоследствии будет использоваться множеством разработчиков по всему миру.

Требования к претенденту. Навыки в программировании на C/C++, желание разбираться в алгоритмах обработки изображений, желание разрабатывать высокопроизводительный код. Опыт оптимизации приветствуется.

Задача 7. Разработка стабильного, точного и простого в использовании инструмента для измерения производительности математических функций

Описание. При разработке ПО часто возникает задача измерения производительности. Для получения точных и стабильных результатов нередко требуется применять сложные подходы и различные функции для таймирования. Также нужно максимально изолировать результаты от внешних факторов: влияния ОС, состояния памяти на момент исполнения и т.д.

Задача заключается в разработке простого, простого в использовании инструмента для замера производительности, который позволил бы получать достаточно стабильные и точные результаты.

Требования к претенденту. Навыки в программировании на C/C++, понимание устройства и принципов работы ОС. Навыки работы со скриптовыми языками (Python, bash) приветствуются.

Задача 8. Продуктизация приложения для визуализации результатов обработки изображений с помощью функций из библиотеки Intel® IPP

Описание. Библиотека Intel® IPP содержит широкий набор функций обработки изображений, таких как фильтрация, геометрические преобразования, преобразования цветового пространства, и т.д. Задача состоит в улучшении и расширении текущего демо-приложения для визуализации результатов обработки изображений в интерактивном режиме.

Требования к претенденту. Навыки программирования на C/C++, приветствуются знания алгоритмов обработки изображений. Опыт работы с графическим пакетом QT.

Задача 9. Интеграция Intel® DAAL в TensorFlow*

Описание. Intel® DAAL (<https://software.intel.com/en-us/intel-daal>) – это библиотека оптимизированных алгоритмов машинного обучения и анализа данных, которая позволяет решать большие высокопроизводительные вычислительные задачи. Фреймворк TensorFlow является самым популярным в области машинного обучения. Необходимо интегрировать библиотеку Intel® DAAL в фреймворк TensorFlow.

Требования к претенденту.

Умение программировать на C/C++.

Хороший математический аппарат (численные методы, теория вероятностей / математическая статистика).

Опыт работы в ОС Linux – плюс.

Знакомство с TensorFlow и Python – плюс.

Задача 10. Разработка процесса поставок программных продуктов по запросам заказчиков вне расписания в соответствии с инженерными и продуктовыми требованиями

Описание. Задача относится к области управления проектами. Для создания процесса, который отвечает требованиям выпуска программных продуктов, необходимо:

- изучить текущий процесс выпуска главных версий продуктов,

- проанализировать требования к поставкам, которые выпускаются по запросу от заказчиков,

- выявить различия между типами поставок,

- разработать процесс, соответствующий каждому типу поставки продуктов. Задача подойдёт кандидатам в том числе нетехническим специальностям.

Для получения необходимой для решения задачи информации придётся много общаться с командами разработчиков, инженерами, маркетологами, юридическим отделом как на командных встречах, по телефону, так и в переписке.

Требования к претенденту.

Уровень владения англ языком intermediate/upper intermediate и выше.

Умение пользоваться приложениями Microsoft Office для обмена, быстрого сбора и обработки всех видов информации.

Дополнительный плюс - знание рынка программных продуктов, прежде всего тех, которые выпускает компания Intel.

Задача 11. Расширение функциональности компоненты Flow Graph библиотеки Intel® Threading Building Blocks (Intel® TBB)

Описание. Компонента Flow Graph – инструмент, позволяющий представить приложение в виде графа, передающего сообщения между своими узлами, исполняющими поставленные задачи.

Задача заключается в исследовании новой функциональности для Flow Graph на основе реальных пользовательских сценариев.

Например, возможность динамического задания количества входных сообщений для join узла (более подробную информацию можно найти по адресу: <https://software.intel.com/en-us/node/506236>).

Необходимо рассмотреть возможные варианты реализации, написать тесты корректности и производительности для реализованного алгоритма и внедрить полученные результаты в библиотеку Intel® TBB.

Требования к претенденту. Знание C++ (желательно C++11/14)

Шаблонное программирование

Многопоточное программирование

Английский на уровне чтения технической документации

Задача 12. Реализация поддержки возможностей последних версий стандарта C++ в Intel® TBB

Описание. Intel® Threading Building Blocks (Intel® TBB) — кроссплатформенная библиотека шаблонов C++, разработанная для параллельного программирования. Библиотекой поддерживаются операционные системы Windows*, Linux*, macOS*, и C++ компиляторы от GNU*, LLVM*, Intel®, Microsoft*.

Intel® TBB предоставляет шаблонные алгоритмы, потокобезопасные контейнеры, синхронизации, аллокаторы и т.п. и позволяет пользователям упростить разработку параллельных программ.

Необходимо ознакомиться с функционалом библиотеки Intel® TBB и последними версиями стандарта C++ и применить новые возможности стандарта C++ в библиотеке.

Требования к претенденту.

Знание C++ (желательно C++11/14/17).

Шаблонное программирование.

STL.

Многопоточное программирование.

Английский на уровне чтения технической документации.

Задача 13. Интеграционное тестирование алгоритмов классического машинного обучения в Intel® DAAL

Описание. Intel® Data Analytics Acceleration Library (<https://software.intel.com/en-us/intel-daal>) - это продукт, который представляет из себя набор блоков, оптимизированных для работы на всех платформах Intel, которые можно использовать на всех стадиях анализа данных и машинного обучения.

Задача: переиспользовать тестовые сценарии для классических алгоритмов машинного обучения (таких, как Gradient Boosting, SVM, K-means и т.д.) из других популярных

фреймворков - Scikit-Learn, TensorFlow, XGBoost, включить их в регулярное тестирование DAAL с целью более простой интеграции библиотеки в перечисленные фреймворки.

Требования к претенденту.

Умение программировать на C/C++ и Python.

Знакомство с алгоритмами машинного обучения.

Опыт работы с ОС Linux.

Знания в областях, связанных с процессом тестирования, будут дополнительным плюсом.

Задача 14. Алгоритмическая оптимизация нейронных сетей

Описание. Задача предполагает знакомство со сверточными нейронными сетями (CNN) и их использованием применительно к алгоритмам компьютерного зрения. Необходимо изучить алгоритм динамического прунинга сверточных нейронных сетей и применить его к готовым архитектурам, разработанным внутри компании, с целью оптимизации количества вычислений и ускорения работы данных сетей на аппаратном обеспечении компании Intel. Помимо использования популярных фреймворков для глубокого обучения, таких как Caffe, Pytorch, в задаче предполагается применение Intel® Deep Learning Deployment Toolkit со специально разработанным расширением для выполнения оптимизированных CNN моделей и замера их производительности. Результатом работы будет набор экспериментов, показывающий эффективность применения алгоритма динамического прунинга к различным архитектурам (топологиям) сетей.

Требования к претенденту.

Знакомство с алгоритмами машинного обучения и обработки изображений.

Опыт программирования на C++, Python.

Приветствуются знания и опыт работы с глубоким обучением.

Задача 15. Исследование и разработка библиотеки глубокого обучения при решении задачи семантической сегментации

Описание. Проект посвящен исследованию возможностей нейронных сетей при решении задачи семантической сегментации. Необходимо изучить современные публикации по тематике семантической сегментации, написать необходимое расширение для библиотеки тренировки нейронных сетей с использованием TensorFlow и выполнить тренировку модели, решающую задачу сегментации. Разработанное расширение и натренированная модель будут включены в разрабатываемую библиотеку, которую планируется выложить в OpenSource.

Требования к претенденту.

Хорошие знания python.

Общее представление о машинном обучении, нейронных сетях и глубоком обучении.

Задача 16. Реализация и сравнение стратегий распределения вычислений и данных на аппаратной платформе Intel® Movidius™ в задачах вычисления моделей глубокого обучения

Описание. Сверточные нейронные сети являются приоритетным направлением современного компьютерного зрения.

Платформа Intel® Movidius™ является аппаратным ускорителем для задач, использующих глубокое обучение. Эффективное использование платформы требует оптимального разбиения вычислений и передачу данных между независимыми вычислительными модулями.

В задачу входит написание и сравнение алгоритмов распределения вычислений на примере базовых операций, используемых в глубоком обучении.

Требования к претенденту. C/C++, Linux*

Задача 17. **Портирование алгоритма компьютерного зрения на ПЛИС**

Описание. Алгоритмы компьютерного зрения и, в особенности, основанные на технологии глубокого обучения требуют значительных вычислительных ресурсов. Специализированные акселераторы позволяют на системном уровне улучшить метрику: производительность / ватт / доллар. Для использования преимуществ акселератора предлагается подготовить приложение, демонстрирующее преимущества выполнения алгоритма компьютерного зрения на ПЛИС

Требования к претенденту. C/C++

Задача 18. **Внедрение алгоритмов компьютерного зрения в приложение**

Описание. Требуется внедрить несколько алгоритмов компьютерного зрения (детекторы объектов и их атрибутов, базирующиеся на нейронных сетях) в конвейер обработки данных приложения, необходимо предусмотреть задание параметров алгоритмов через графический интерфейс пользователя и визуализацию обнаруженных объектов и их признаков.

Требования к претенденту. Хорошее знание C++, опыт разработки приложений с графическим интерфейсом для ОС Windows*/Linux*. Предпочтительно знание фреймворка Qt, знакомство с системой контроля версий GitHub.

Задача 19. **Разработка структур данных для систем с гетерогенной памятью**

Описание. Подсистема памяти современных серверных решений не является однородной. Технология Intel 3D XPoint позволяет создавать чипы памяти большей емкости по сравнению с текущими решениями DRAM. Целью задачи является создание структур данных для систем с гетерогенной памятью. Структуры данных должны управлять размещением данных между различными типами системной памяти. Элементы структур данных должны перемещаться во время выполнения между различными типами памяти на основе паттерна доступа к данным.

Требования к претенденту. C++, структуры данных и алгоритмы, метапрограммирование, ООП.

Задача 20. **Модифицировать структуры данных библиотеки Intel® TBB для использования энергонезависимой памяти**

Описание. Постоянная память (English: persistent memory) - это тип энергонезависимой памяти, которая может адресоваться с гранулярностью байт (как и DRAM), и сохраняет свое состояние в процессе перезапуска и перезагрузки системы. Персистентная память доступна для приложений через механизм отображения файлов в адресное пространство процесса. Новый тип памяти позволяет создавать структуры данных со временем жизни, превышающем время жизни процесса. Но существующие структуры данных (в STL или Intel® TBB), предназначенные для энергозависимой памяти, не могут использоваться для персистентной памяти и должны быть портированы. Структуры данных, предназначенные для постоянной памяти, должны использовать специальный аллокатор, который может выделять память из определенного файла, отображенного в виртуальную память процесса. Кроме того, обычные указатели не могут быть использованы, потому что после перезапуска процесса регион персистентной памяти может быть отображен в другие виртуальные адреса. В рамках данной задачи планируется переработать структуры данных библиотеки Intel® TBB для поддержки персистентной памяти.

Требования к претенденту. C++, структуры данных и алгоритмы, метапрограммирование, ООП

Задача 21. **Микроархитектурная оптимизация приложений для высокопроизводительных систем**

Описание. Подготовка HPC-приложений для анализа: сборка, запуск, поиск узких мест в профиле и внедрение архитектурных оптимизаций. Список целевых приложений: OpenFOAM (симуляция гидрогазодинамических процессов), COSMO (вычислительная химия), miniFE (солвер метода конечных элементов).

Требования к претенденту. Знание архитектуры x86 начальный уровень, C/C++ средний уровень, Linux* обычный пользователь, знания MPI/OpenMP/компилятора как дополнительный плюс

Задача 22. Визуализатор для энкодеров Intel®Media SDK

Описание. Требуется создать ПО для визуализации результатов сжатия видео посредством Intel® Media SDK, а также для сравнения полученного результата с оригинальным несжатым видео. ПО должно показывать, как разные настройки сжатия влияют на результат, и показывать различия визуально.

Требования к претенденту. Хорошее знание C++, опыт программирования для ОС Windows*/Linux*. Предпочтительно знание фреймворка QT.

Задача 23. Разработка модуля для визуализации 3D-модели окружающей среды автономного автомобиля с использованием Web-технологий

Описание. Необходимо проработать дизайн и имплементацию Web-модуля для визуализации 3D-модели окружающей среды автономного автомобиля с использованием современных Web-технологий. Модель окружающей среды включает в себя окружающие автомобили, полосы движения, светофоры, пешеходы и т.д. Визуализация модели окружающей среды необходима для тестирования и изучения работы алгоритмов построения модели окружающей среды, на которую ориентируются автономные автомобили при принятии решений. Данный модуль будет использоваться в сложной системе по визуализации и разметке данных с разнообразных сенсоров и использованием кластерных технологий, поэтому модуль должен поддерживать множество источников разнообразных данных.

Требования к претенденту. OpenGL, WebGL, Javascript, HTML5, 3D Web-фреймворки, такие как three.js

Задача 24. Анализ производительности параллельных алгоритмов стандартной библиотеки C++ (Parallel STL)

Описание. Parallel STL (доступная в версии C++17) - это параллельные версии алгоритмов стандартной библиотеки шаблонов. Необходимо сделать анализ производительности параллельной / векторной реализации алгоритмов Parallel STL, базирующихся на библиотеке Intel® TBB и возможностях векторизации компилятора Intel.

Требования к претенденту. Знание: C++11, STL, обобщенное программирование.

Задача 25. Осуществление поддержки нейронных сетей в формате хранения ONNX в OpenCV.

Описание. Разные инструменты для тренировки нейронных сетей используют собственные форматы для хранения готовых топологий. Конвертация между ними часто возможна лишь вручную и требует глубоких знаний особенностей работы каждого из этих инструментов. ONNX (<http://onnx.ai>) - новый стандарт хранения глубоких сетей, поддерживаемый многими большими компаниями, включая Intel, Amazon, Microsoft и т.д., который призван решить эту проблему. В частности, новая версия популярного инструмента для тренировки сетей Caffe v2 поддерживает ONNX формат.

Задачи:

- Изучить, какие известные топологии представимы в формате ONNX.
- Создать интерфейс для чтения моделей из формата хранения ONNX во внутренний формат представления сетей в OpenCV.
- При необходимости добавить новые слои или улучшить существующие для поддержки новых моделей.
- Написать регрессионные тесты как для готовых натренированных сетей, так и для точечных сгенерированных однослойных моделей для проверки корректности работы загрузчика.

- Написать примеры использования в OpenCV сетей для решения наиболее интересных задач.

Требования к претенденту. Очень хорошие навыки в программировании на C++ обязательны. Базовые знания/опыт в чтении форматов данных и в устройстве глубинных сетей желательны.

Задача 26. Исследование и разработка алгоритмов глубокого обучения для задачи детектирования объектов

Описание. Задача связана с алгоритмами глубокого обучения, основанными на сверточных нейронных сетях для задачи детектирования объектов.

Задача состоит в исследовании и разработке различных state-of-the-art моделей для задачи детектирования объектов с помощью open-source библиотек глубокого обучения (Caffe, TensorFlow, PyTorch), которые впоследствии будут использоваться внутри ICV команды. Также ожидается не только разработка качественных моделей, но и внедрение получившихся архитектур с помощью инструментов Model Optimizer, Inference Engine для различных аппаратных платформ.

Требования к претенденту. Хорошее понимание алгоритмов машинного обучения, опыт написания программ на C++/Python.

Опыт работы с алгоритмами глубокого обучения для задач компьютерного зрения будет плюсом.

Задача 27. Оптимизация OpenCV для новых способов организации вычислений

Описание. OpenCV эволюционирует, и частью этой эволюции является исследование новых способов организации вычислений. Трансформации над потоком выполнения программы позволяют применять параллелизм, локальность данных, уменьшить потребление памяти, а также задействовать гетерогенные вычисления. Одним из способов полуавтоматически добиться перечисленных качеств является переход к графовой модели представления алгоритма. Строительными блоками в такой модели становятся ядра - примитивные функции компьютерного зрения, написанные особым образом.

Задачи:

- Изучить, как разрабатываются ядра для графовой модели OpenCV;
- Реализовать набор ядер для одной из платформ (Intel, OpenCL, Movidius, Halide);
- Научиться основам оптимизации с использованием компилятора.

Требования к претенденту. Обязательно: хорошие навыки в программировании на C++.

Желательно: начальные навыки программной оптимизации

Задача 28. Интеграция кодека AV1 в кроссплатформенный свободный продукт для конвертации форматов видеофайлов HandBrake

Описание. AOMedia Video 1, (AV1) – свободный открытый формат видеокодирования, спроектированный для передачи видео по сети. Он разрабатывается основанным в 2015 года Alliance for Open Media, объединяющим компании-производители железа, провайдеров видео-контента и разработчиков веб-браузеров.

HandBrake – бесплатный открытый транскодер видеофайлов с очень большой пользовательской аудиторией.

Задача – добавить поддержку AV1 в HandBrake.

Требования к претенденту.

Обязательные требования: знание языка C, общее понимание принципов работы и интерфейса видеокодеков. Опыт участия в проекте с открытым кодом. Знание английского, достаточное для чтения технической документации. Умение программировать в среде Windows* и/или Linux*. Способность к самостоятельной работе.

Желательные навыки: более глубокое знание в области кодирования видео, опыт работы с git.

Задача 29. Анализ изображений с целью извлечения мультиспектральных данных

Описание. Задача состоит в получении реконструированных узкополосных инфракрасных изображений путем анализа данных RGB камеры и/или широкополосной IR камеры. Необходимо провести исследование существующих алгоритмов и создать ПО, реализующее данные алгоритмы.

Требования к претенденту. Хорошее знание C++, опыт использования OpenCV. Предпочтителен опыт работы с фреймворками для сверточных нейронных сетей, а также знание системы MathLab.

Задача 30. Разработка систем для автоматизации тренировки и тестирования алгоритмов глубокого обучения на кластере

Описание. Для тренировки и тестирования алгоритмов глубокого обучения нужно иметь отказоустойчивую и масштабируемую систему менеджмента и распределения ресурсов кластера. Существует несколько Open Source проектов, которые пытаются реализовать данную инфраструктуру. В рамках настоящего проекта нужно будет выбрать одно из существующих решений, доработать и внедрить его в команде ICV. Конечной целью является ускорение процесса тренировки алгоритмов глубокого обучения за счет более эффективного использования вычислительных мощностей.

Требования к претенденту. Хорошие знания Python 3.x, C/C++. Умение работать с Git. Опыт разработки на Linux*. Опыт со SLURM (<https://slurm.schedmd.com/>) или PAI (<https://github.com/Microsoft/pai>) будет плюсом. Общее представление о машинном обучении, нейронных сетях и глубоком обучении.

Задача 31. Интеграция инструмента контроля корректности видеокодирования ASG (Artificial Stream Generator) в open-source экосистему Intel® Media Server Studio for Linux*

Описание. Intel® Media Server Studio for Linux* – ведущее решение для центров обработки данных видеостриминговых компаний, воплощающие современные принципы и технологии эффективной обработки медиаконтента. Необходимо создать инфраструктуру для использования инструмента ASG (Artificial Stream Generator) в тестовой системе растущего open-source-сообщества Intel® MediaSDK (<https://github.com/Intel-Media-SDK/MediaSDK>).

Требования к претенденту. Базовые знания Python, Perl и опыт работы с OS Linux*. Дополнительным плюсом будут знания C/C++ и основ алгоритмов видеокодирования.

Задача 32. Стенд для демонстрации возможностей процессоров Intel в приложениях реального времени

Описание. Спроектировать и реализовать программно-аппаратный макет для демонстрации возможностей создания realtime приложений с использованием средств разработки Intel, включая возможности координированной во времени обработки событий. Стенд представляет собой систему управления с обратной связью с жесткими временными требованиями к реализации управляющих воздействий с сенсоров.

Требования к претенденту. Низкоуровневое программирование (C, Assembler), OS Linux.