

Баялы А.Т.¹, Жунисов Н.М.², Назарбек Т.С.³, Давлетова В.М.⁴

¹аға оқытушы, Қожас Ахмет Ясауи атындағы Халықаралық қазақ-түрік университеті, (Қазақстан, Түркістан қ.), e-mail: azimkhan.bayaly@ayu.edu.kz

²PhD, Қожас Ахмет Ясауи атындағы Халықаралық қазақ-түрік университеті, (Қазақстан, Түркістан қ.), e-mail: nurseit.zhunissov@ayu.edu.kz

³магистр оқытушы, Қожас Ахмет Ясауи атындағы Халықаралық қазақ-түрік университеті, (Қазақстан, Түркістан қ.), e-mail: Tolkin.Nazarbek@ayu.edu.kz

⁴магистр оқытушы, Қожас Ахмет Ясауи атындағы Халықаралық қазақ-түрік университеті, (Қазақстан, Түркістан қ.), e-mail: venera.davletova@ayu.edu.kz

ПАРАЛЛЕЛЬ АЛГОРИТМДІ ПАЙДАЛАНА ОТЫРЫП СУРЕТТЕРДІ ӨНДЕУ ӘДІСТЕРІ IMAGE PROCESSING METHODS USING A PARALLEL ALGORITHM МЕТОДЫ ОБРАБОТКИ ИЗОБРАЖЕНИЙ С ИСПОЛЬЗОВАНИЕМ ПАРАЛЛЕЛЬНОГО АЛГОРИТМА

Аңдатпа. Бүгінгі таңда суреттерді өңдеу түрлі ғылыми және технологиялық салаларда маңызды орын алады. Деректер көлемінің ұлғаюы мен ақпаратты жедел өңдеу қажеттілігі өңдеу әдістерін жетілдіруге итермелейді. Осы тұрғыда қатар жүргізілетін алгоритмдерді қолдану суреттерді өңдеудің тиімділігін арттырудың басты тәсілдерінің біріне айналып отыр. Бұл жұмыс параллель алгоритм кескін өңдеу әдістерін зерттеуге арналған. Жұмыс барысында кескіндерді өңдеудің негізгі қағидалары қарастырылады. Сонымен қатар, бұл әдістер тапсырмаларды орындау жылдамдығын арттырып, есептеу ресурстарын үнемді пайдалануға мүмкіндік береді. Параллель алгоритм өңдеудің негіздері мен оның артықшылықтары жан-жақты сипатталады. Әртүрлі сәулеттермен жұмыс істейтін қатар жүргізілетін есептеулерге ерекше назар аударылады. Әсіресе бірнеше технологияны қамтитын алгоритмдер мысал ретінде көрсетіледі. Параллель алгоритм тәсіл арқылы кескіннің әртүрлі бөліктеріне бір мезетте бірнеше сүзгілерді қолдануға болады. Бұл әдіс жүйенің жалпы өнімділігін көтеріп, өңдеуге кететін уақытты азайтады. Сонымен бірге, көп ядролы процессорлар мен графикалық процессорлар сияқты құрылғыларды тиімді пайдалану жоғары сапалы кескін өңдеуді қамтамасыз етеді. Қолданылатын әрбір сүзгінің өзіндік ерекшеліктері бар және олар нақты мақсаттарға бағытталған. Параллель алгоритм әдістердің арқасында бұл сүзгілерді тез әрі сапалы қолдану мүмкіндігі туады. Нақты уақыт режиміндегі бейнелерді өңдеуде үлкен рөл атқарады. Жоғары тиімділікке ие қатар жүргізілетін алгоритмдер кескіндерді өңдеудің нәтижесін едәуір жақсартады.

Негізгі сөздер: Параллель алгоритм, кітапхана, OpenCV кітапхана, интерфейс, сүзгі, фильтр, кескін.

Abstract. Today, image processing occupies an important place in various scientific and technological fields. The increase in data volume and the need for rapid information processing are pushing for improved processing methods. In this context, the use of parallel algorithms is becoming one of the main ways to increase the efficiency of image processing. This work is devoted to the study of image processing methods that are carried out in parallel. In the course of the work, the basic principles of image processing are considered. In addition, these methods allow you to increase the speed of tasks and use computing resources economically. The basics of parallel processing and its advantages are described in detail. Special attention is paid to parallel computing working with different architectures. Algorithms involving several technologies are especially shown as an example. With a parallel approach, you can apply multiple filters to different parts of the image at the same time. This method improves the overall performance of the system and reduces the time spent on processing. At the same time, the efficient use of devices such as multi-core processors and graphics processors ensures high-quality image processing. Each filter used has its own characteristics and is aimed at specific purposes. Thanks to parallel methods, it becomes possible to quickly and efficiently apply these filters. It plays a big role in real-time video editing. Parallel algorithms with high efficiency significantly improve the result of image processing.

Keywords: Parallel algorithm, library, OpenCV library, interface, filter, filter, image.

Аннотация. Сегодня обработка изображений занимает важное место в различных научных и технологических областях. Увеличение объема данных и необходимость оперативной обработки информации подталкивают к совершенствованию методов обработки. В этом контексте применение параллельных алгоритмов становится одним из главных способов повышения эффективности обработки изображений. Данная работа посвящена изучению методов обработки изображений, которые проводятся параллельно. В процессе работы рассматриваются основные принципы обработки изображений. Кроме того, эти методы позволяют увеличить скорость выполнения задач и экономно использовать вычислительные ресурсы.

Подробно описаны основы параллельной обработки и ее преимущества. Особое внимание уделяется параллельным вычислениям, работающим с различными архитектурами. Особенно в качестве примера показаны алгоритмы, включающие несколько технологий. При параллельном подходе можно одновременно применять несколько фильтров к разным частям изображения. Этот метод повышает общую производительность системы и сокращает время, затрачиваемое на обработку. В то же время эффективное использование таких устройств, как многоядерные процессоры и графические процессоры, обеспечивает высококачественную обработку изображений. Каждый используемый фильтр имеет свои особенности и нацелен на конкретные цели. Благодаря параллельным методам появляется возможность быстрого и качественного применения этих фильтров. Играет большую роль в редактировании видео в реальном времени. Параллельные алгоритмы, обладающие высокой эффективностью, значительно улучшают результат обработки изображений.

***Ключевые слова.** Параллельный алгоритм, библиотека, OpenCV библиотека, интерфейс, фильтр, фильтр, изображение.*

Кіріспе

Кескіндерді өңдеу – бұл визуалды мәліметтерді талдау және түрлендіру үдерістерін қамтитын ақпараттық технологиялар саласындағы маңызды бағыттардың бірі. Қазіргі таңда бұл технология қауіпсіздік, көлік, ойын-сауық сияқты көптеген салаларда кеңінен қолданылып келеді. Мысал ретінде медицина саласын қарастырайық: рентген суреттері мен МРТ кескіндерін өңдеу арқылы түрлі ауруларды анықтау және диагноз қою мүмкін болады. Қауіпсіздік саласында бейнебақылау жүйелері күмәнді әрекеттерді анықтау мақсатында арнайы алгоритмдерге сүйенеді. Сонымен қатар, компьютерлік көру жүйелері автономды көліктердің негізгі құрамдас бөлігі ретінде қызмет атқарып, көліктің қоршаған ортаны қабылдап, сараптауына мүмкіндік береді[1].

Параллель алгоритм – өңдеу үдерістерін бірнеше ядро немесе процессор арасында бөлісу арқылы орындалатын тәсіл. Мұндай әдіс өңдеудің тиімділігін арттырып, уақытты үнемдеуге мүмкіндік береді. Себебі бір уақытта суреттің бірнеше аймағын қатар өңдеуге болады. Мысал ретінде кескінді бірнеше сегментке бөліп, оларды қатар жүргізілетін түрде талдау арқылы жалпы өңдеу уақытын бірнеше есе қысқартуға болады. Сонымен қатар, бұл әдіс есептеу ресурстарын тиімді пайдалану мүмкіндігін береді, әсіресе есептеу қуаты шектеулі құрылғыларда өте өзекті болып табылады.

Бұл зерттеудің мақсаты параллель алгоритмдер кескінді өңдеу жылдамдығын арттыру. Визуализация технологияларының қазіргі дамуының маңызды бөлігіне айналдыру.

Параллель алгоритмдер – бұл есептеу процестерін қатар жүргізілетін түрде, яғни бір уақытта бірнеше кіші бөлшектерге бөліп орындайтын алгоритмдер. Бұл тәсілдің негізгі мақсаты – жүйенің ресурстарын тиімді пайдалану және есептеудің жалпы уақытын азайту. Бұл технологияны іске асыру үшін көп ядролы процессорлар мен графикалық процессорлар (GPU) кеңінен қолданылады, өйткені олар бірнеше есептеу операцияларын бір мезгілде орындай алады. Мысалы, қазіргі заманғы компьютерлерде бірнеше ядролар немесе графикалық процессорлар (GPU) бар. Параллель алгоритмдер осы мүмкіндікті пайдаланып, әрбір ядро немесе графикалық процессор бөлек тапсырманы орындайды. Бұл әдіс әсіресе үлкен деректерді өңдеу, күрделі есептеулер мен бейнемазмұнды өңдеуде өте тиімді болып табылады. Ал енді, бейнелерді өңдеуде қатар жүргізілетін алгоритмдердің артықшылықтарын қарастырайық[1]. Күрделі бейнелерді өңдеу кезінде көптеген есептеулер мен математикалық операциялар орындалады, және егер әрбір операцияны бірінен соң бірі орындаса, бұл уақытты көп алады. Параллель алгоритмдер бұл процессті бірнеше бөлікке бөліп, әрбір бөлікке арнайы ресурстар бөлінгенде, бүкіл процесс әлдеқайда жылдам орындалады:

1. Бейнелерді сүзгілеу – бірнеше түрлі сүзгілер қолданылатын кезде, әрбір сүзгіні параллельді түрде орындауға болады.
2. Бейне аналитика – бейнемазмұндағы объектілерді тану немесе қозғалысты бақылау сияқты тапсырмаларды бірнеше ядро орындағанда әлдеқайда тез өңдеуге болады.

3. Анимация және жасанды интеллект – күрделі есептеулер мен модельдерді іске асыруда қатар жүргізу тиімділік береді.

Параллель әдісі көптеген қазіргі заманғы технологияларда қолданылуда, өйткені ол жүйенің өнімділігін арттырып, пайдаланушының күту уақытын азайтады. Бұл тәсіл әсіресе көп ядролы графика мен есептеу құрылғыларын қолдайтын қолданбалар үшін өте тиімді [1].

Grama, Gupta, Karupis және Kumar (2003) еңбегі болып табылатын "Introduction to Parallel Computing" кітабы Параллель есептеулер саласындағы негізгі әдебиеттердің бірі ретінде кеңінен танылады. Бұл еңбекте параллель жүйелердің құрылымы, алгоритмдерді параллель түрде ұйымдастыру қағидалары, есептеу модельдері мен өнімділікті арттыру тәсілдері жан-жақты сипатталған. Аталған жұмыс ресурсты көп қажет ететін есептер мен кескіндерді өңдеу салалары үшін тиімді алгоритмдерді құруға теориялық негіз қалыптастырған.

Параллель кескін өңдеудің қолданбалы аспектілері.

С. В. Егіздер, В. Г. Меледин және В. А. Павлов (2017) өз зерттеу жұмыстарында фазалық кескіндерді жоғары жылдамдықпен өңдеу мәселесіне тоқталады. Зерттеуде параллель есептеу құрылымдарын, соның ішінде мультипроцессорлық жүйелерді пайдалану арқылы фазалық карталарды талдау жылдамдығын едәуір арттыру мүмкін екендігі көрсетілген. Бұл тәсілдер техникалық көру мәселелері үшін аса маңызды.

Клебан Е. Г. (2020) өз еңбегінде кескіндер мен бейне ағындарын өңдеуге арналған нақты параллель алгоритмдерді ұсынады. Автор OpenMP және CUDA технологияларының көмегімен бейнені сүзу, шекараларды анықтау және деректерді қысу сияқты классикалық және заманауи әдістерді егжей-тегжейлі қарастырады.

Sandipan Dey өзінің "Python Image Processing Cookbook" (2020) атты еңбегінде Python тіліндегі заманауи кітапханалар арқылы кескіндерді өңдеуге арналған практикалық тәсілдерді ұсынады. Кітаптың қолданбалы мазмұнына қарамастан, автор көп ағынды және GPU-мен жұмыс істеу арқылы өнімділікті арттыру мәселелерін де талдайды.

Richard Szeliski "Computer Vision: Algorithms and Applications" (2020) атты еңбегінде компьютерлік көру алгоритмдерін жүзеге асыру мәселелерін қозғайды. Ол параллель алгоритмдер мен үлкен көлемдегі деректермен нақты уақыт режимінде жұмыс істеу тәсілдерін егжей-тегжейлі баяндайды.

Қазақстандық зерттеулер аясында.

Қазақстандық ғылыми ортада да Параллель өңдеу тақырыбы белсенді түрде зерттелуде. Мәселен, Смайылов Ж. (2017) "Мәліметтерді параллель өңдеудегі тиімді әділет" атты еңбегінде үлкен деректер массивін талдауға арналған Параллель есептеу әдістерін ұсынады.

Базарбаев С. (2018) және Әбдірахманов Н. (2021) өз еңбектерінде бейнебақылау жүйелері мен медициналық диагноз қою салаларында параллель алгоритмдерді қолданудың нәтижелілігін көрсетеді.

Жарылқасын Е. (2022) кескіндерді параллель сұрыптаудың түрлі әдістерін ұсына отырып, өңдеу жылдамдығы мен ерекшеліктерді анықтау дәлдігін арттыру жолдарын қарастырады.

Нурпейсова А.А. (2021) Параллель өңдеу тәсілдерін инновациялық өнімдерді құрастыру барысында модельдеу үдерістерінде қалай қолдануға болатынын зерттейді.

Елешова Ш. Д. және оның әріптестері (2021) мобильді құрылғыларға арналған электрондық оқулықтарды әзірлеу кезінде есептеу ресурстарын тиімді үлестіру мәселесіне назар аударады.

Жасанды интеллект пен прокторинг бағытында А. Шаушенова және т. б. (2022), сонымен қатар А. А. Мырзамуратова мен О. Г. Жаңабайқызы (2024) параллель өңдеуді сөйлеуді тануға арналған нейрондық желілер негізінде қолдану әдістерін зерттейді.

Бауыржанұлы Б. және оның ғылыми ұжымы (2021) биометриялық сәйкестендірудің ашық жүйелерге арналған алгоритмдерін ұсына отырып, бұл есептеулердің жоғары өнімді параллель тәсілдерді қажет ететінін атап өтеді.

Қорытындылай келе, әдебиеттерді саралау көрсеткендей, параллель кескін өңдеу - бұл тек есептеу саласының теориялық негіздеріне сүйенетін емес, сонымен қатар медицина, қауіпсіздік, білім және инженерия салаларындағы нақты мәселелерді шешуге бағытталған серпінді дамып жатқан ғылыми бағыт. Шетелдік және отандық зерттеулер бұл бағыттың кең ауқымды қолдану мүмкіндігін және жоғары ғылыми-практикалық маңызын дәлелдейді.

Зерттеу әдістері

Бұл мақалада Параллель алгоритмдердің кескінді өңдеудегі артықшылықтарын қолдана отырып, өңдеудің жылдамдығын арттыруға мүмкіндік беретін тиімді шешімдер қарастырылады. Ауқымды кескіндерді өңдеуде жоғары өнімділікке қол жеткізу үшін және нақты уақыттағы бейнелерді өңдеу үшін қатар жүргізу өте пайдалы тәсіл болып табылады. Бұл әдіс уақытты үнемдеп, үлкен көлемдегі деректермен жұмыс істегенде өңдеудің тиімділігін арттырады [2].

Жұмыстың негізгі мақсаты – Python бағдарламасында кескінді өңдеуге арналған бағдарламаны құру. Бұл бағдарлама OpenCV кітапханасы мен Tkinter қосымшасын пайдаланып, қолданушыларға суреттерді әртүрлі сүзгілер арқылы өзгертуге мүмкіндік береді. Мысалы, Гаусс сүзгісі, шекараны анықтау немесе суреттің сұр реңкке түсу сияқты әртүрлі өңдеу амалдары ұсынылады.

Параллель өңдеу әдісі бағдарламаның тиімділігі мен жылдамдығын арттырады. Бұл әдіс бойынша кескін алдын ала төрт бөлікке бөлінеді, ал әрбір бөлік бір уақытта жеке ағындармен өңделеді. Әрбір ағын өзінің бөлігімен жұмыс істеп, нәтижесінде бүкіл кескінді өңдеу әлдеқайда жылдам орындалады.

Бұл тәсіл қолданушылардың қажеттіліктерін толық қанағаттандыруға бағытталған, себебі әртүрлі сүзгілерді қолдана отырып, өңдеу жылдамдығы мен өнімділігі жоғары болады [3].

```
1 import cv2
2 import numpy as np
3 import threading
4 import tkinter as tk
5 from tkinter import filedialog, Label, Button
6 from PIL import Image, ImageTk
```

Сурет 1. Параллель өңдеу құралдарын шақыру

OpenCV (Open Source Computer Vision Library) - бұл суреттер мен бейнелерді өңдеуге арналған қуатты кітапхана. Ол графикалық операцияларды орындау үшін арнайы құралдар мен әдістерді ұсынады, мысалы, суреттерді сүзу, шекараны анықтау, сегменттеу және тағы басқа көптеген өңдеу әдістері. OpenCV кітапханасы кескіндермен манипуляция жасауға және оларды әртүрлі тәсілдермен өңдеуге арналған бай мүмкіндіктерге ие [4].

OpenCV-нің ең негізгі артықшылықтарының бірі - оның әртүрлі сүзгілерді қолдану мүмкіндігі:

1. Гаусс сүзгісі - суреттің шуын азайту үшін қолданылады.
2. Шеттеріндегі айқындықты арттыру - кескіннің шекараларын анықтап, қоюлығын жақсартады.
3. Сұр реңк сүзгісі - кескінді тек сұр реңктерде көрсетеді, бұл кейбір бейнемазмұнды өңдеу кезінде қажет болуы мүмкін.

Параллель өңдеу әдісін қолдану арқылы OpenCV барлық осы сүзгілерді бірнеше бөлікке бөлінген кескіндерге параллельді түрде қолдануға мүмкіндік береді. Кескінді бірнеше ағындарға бөлу арқылы оларды бір уақытта өңдеуге болады. Бұл өңдеу процесін

жылдамдатуға, әсіресе үлкен кескіндер мен бейнемазмұнды өңдеу кезінде уақытты үнемдеуге мүмкіндік береді.

OpenCV ішкі алгоритмдері C тілінде жазылған, бұл оның жоғары өнімділігін қамтамасыз етеді. Python тілін қолданатын бағдарлама OpenCV-ді қолдану арқылы өңдеу процесін жылдамдатуға көмектеседі. Python-ның қарапайымдылығы мен OpenCV-дің тиімділігі бірлесіп, кескіндерді өңдеуді тиімді әрі жылдам жүзеге асыруға мүмкіндік береді. Бұған қосымша, OpenCV бірнеше деректер пәрмендерін (SIMD — Single Instruction, Multiple Data) қолдайды. Бір уақытта бірнеше деректерді өңдеуге мүмкіндік беретін тәсіл. Мысалы, бір команданы қолдану арқылы бірнеше пиксельдерге бірдей өңдеу амалын бір мезгілде орындауға болады. Бұл үлкен көлемдегі кескіндерді немесе бейнелерді өңдеу кезінде өте маңызды рөл атқарады, себебі оны өңдеу бірнеше ағын арқылы жүзеге асады, бұл жалпы уақытты айтарлықтай қысқартады [5]. Осылайша, OpenCV — бұл кескіндерді өңдеудің жоғары өнімділікті, тиімді және қуатты құралдары бар кітапхана, және қатар жүргізілетін өңдеу тәсілдерін қолданып, кескіндерді жылдам өңдеуге мүмкіндік береді.

NumPy (np): NumPy — математикалық және ғылыми есептеулерді жүргізуде кеңінен қолданылатын қуатты құрал. Бұл кітапхана кескіндерді өңдеу кезінде, оларды кесте түріндегі деректер ретінде қарастыруға мүмкіндік береді. NumPy көмегімен суреттегі әрбір пиксельді массив элементі ретінде қарастырып, олармен сандық амалдарды оңай жүзеге асыруға болады. Бұл кітапхана кескіндерді өңдеудің сапасын арттырып, ресурстарды тиімді пайдалануға жол ашады.

Threading: Параллель ағындармен жұмыс істеуге арналған стандартты Python модулі. Python тіліндегі стандартты threading модулі бірнеше ағындарды бір мезетте іске қосуға мүмкіндік береді. Кескіндерді өңдеуде олардың жекелеген бөліктерін бір уақытта өңдеу қажет болғанда тиімді. Ағындар (жіптер) бір уақытта түрлі есептеулерді орындауға бейімделіп, әрбіреуі кескіннің өзіне тиесілі бөлігін өңдей алады. Нәтижесінде кескін толықтай параллель түрде өңделіп, соңында бір бүтін суретке біріктіріледі. Бұл тәсіл барлық есептеулер біткеннен кейін нәтижелерді қайта біріктіруге жол ашады [6].

Tkinter (tk): Tkinter — Python-да графикалық интерфейс құруға арналған стандартты құрал. Python тіліндегі Tkinter модулі графикалық интерфейстерді жасау үшін жиі қолданылады және ол қолданушы мен бағдарлама арасындағы өзара әрекеттесу элементтерін — мысалы, түймелер мен мәтін белгілерін — оңай жасауға мүмкіндік береді. Бұл модульдің көмегімен суреттерді таңдау, өңдеу және қарау үшін интуитивті әрі қарапайым интерфейс әзірлеуге болады. Пайдаланушыға суретке енгізілген өзгерістерді бірден бақылауға мүмкіндік береді. Виджеттер арқылы өңделген кескіндерді көрсету және өңдеу опцияларын батырмалар арқылы іске қосу қарапайым әрі қолайлы түрде жүзеге асады.

Tkinter - Python-да графикалық пайдаланушы интерфейстерін (GUI) құруға арналған стандартты кітапхана. Tkinter компоненттері арқылы қолданушы интерфейсін жасауда көптеген пайдалы құралдар мен виджеттер бар. Компоненттерді пайдалану арқылы біз графикалық элементтерді оңай және тиімді түрде орналастыра аламыз. Tkinter-дің кейбір негізгі компоненттеріне тоқталып өтейік:

FileDialog — бұл қолданушыға файлды таңдауға мүмкіндік беретін диалогтық терезе. Мысалы, қолданушы суретті таңдағанда, файлды таңдауға арналған терезе ашылады. FileDialog компоненті арқылы қолданушыға белгілі бір файл түрін таңдау немесе файлды ашу әрекеттерін жасауға болады [7].

Мысал ретінде, файлды таңдауға арналған терезе жасауды қарастырайық:

```
python
```

```
Көшіру/өңдеу
```

```
from tkinter import filedialog
```

```
file_path = filedialog.askopenfilename() # Пайдаланушыдан файлды таңдау
```

Label — бұл тек мәтін немесе суреттерді көрсету үшін қолданылатын компонент. Label көмегімен GUI интерфейсінде мәтіндер мен кескіндер көрсетуге болады. Мысалы,

қолданушыға сурет немесе қандай да бір ақпаратты көрсетуде, Label өте тиімді құрал болып табылады. Мысалы:

```
python
Көшіру/өңдеу
from tkinter import Label
label = Label(root, text="Бұл мәтін", font=("Arial", 14))
label.pack()
```

OpenCV-де өңделген суреттерді Tkinter-де көрсету үшін алдымен оларды PIL пішіміне түрлендіру керек. Бұл үшін Image және ImageTk модульдерін қолданамыз. Мысалы, OpenCV арқылы алынған суретті Tkinter терезесінде көрсету үшін төмендегі кодты қарастыруға болады:

Бұл кодта OpenCV арқылы сурет оқылып, PIL-ге түрлендіріліп, Tkinter терезесінде көрсетіледі. PIL кітапханасы суреттерді Tkinter компоненттерінде көрсету үшін қажетті құралдармен қамтамасыз етеді. Сонымен, Tkinter мен PIL кітапханалары бірге жұмыс істегенде, біз суреттермен және файлдармен жұмыс істеуді жеңілдетіп, қолданушы қосымшаларын тиімді түрде құра аламыз. Кескіндерді өңдеуге арналған кодта қолданылатын кітапханалар бірнеше маңызды функцияларды орындайды. Бейнелер мен суреттерді түрлендірумен ғана шектелмей, қатар жүргізілетін ағындарды іске қосу және графикалық интерфейсті ұйымдастыру мүмкіндігін де қамтамасыз етеді. OpenCV, NumPy, Threading және Tkinter сияқты кітапханаларды біріктіре пайдалану өңдеуді жоғары өнімділікпен жүзеге асыруға әрі қолданушыға ыңғайлы басқару ортасын жасауға мүмкіндік береді. Әрбір модульдің кескіндермен жұмыс істеуде өзіндік ерекшелігі мен міндеті бар. Осы кітапханалардың үйлесімді жұмысы қатар жүргізілетін есептеу процестерін жеңілдетіп, жалпы жүйенің тиімділігін арттырады [8].

Параллель өңдеу әдістері кескіндер мен бейнелермен жұмыс істейтін қосымшалардың өнімділігін айтарлықтай арттырып, олардың жылдам әрі тиімді орындалуына ықпал етеді. Бұл технологиялар жүйенің жалпы жұмысын жетілдіруде маңызды орын алады. OpenCV арқылы түрлі фильтрлерді суреттерге қолдануға болады, ал NumPy модулі есептеу процестерін топтап өңдеуге оңтайландырады. Threading модулі бірнеше ағын арқылы бір уақытта түрлі операцияларды орындауға мүмкіндік береді. Tkinter кітапханасы нәтижелерді көрнекі әрі ыңғайлы түрде көрсету үшін графикалық интерфейс құруға жағдай жасайды. Аталған құралдарды біріктіре қолдану арқылы күрделі және ресурсты қажет ететін есептерді жоғары жылдамдықпен әрі дәлдікпен шешуге жол ашылады. Бұл әдіс көлемі үлкен бейне ағындарын немесе суреттерді өңдеуді жеңілдетіп, нақты уақыт режимінде нәтиже ұсынуға және аса күрделі алгоритмдерді тиімді жүзеге асыруға мүмкіндік береді [9].

Кескінді өңдеу – бұл сандық кескіндерге түрлендіру, жақсарту және талдау мақсатында қолданылатын әдістер мен технологиялар жиынтығы. Бұл бағыт көптеген салаларда, соның ішінде медицина, өндіріс, география, қауіпсіздік, өнер және басқа да салаларда кеңінен қолданылады. Кескіндерді өңдеудің негізгі әдістеріне келесі әдістер бар.

Кескінге сүзгісі пайдалану үшін `apple_filter` функциясын пайдаланылады. Бұл функция кескіннің бір бөлігіне таңдамаулы сүзгіні пайдаланады (2-сурет).

```
9 def apply_filter(image_section, filter_type, result, index):
10     """Белгілі бір бөлікте фильтр қолдану."""
11     if filter_type == 'blur':
12         result[index] = cv2.GaussianBlur(image_section, (15, 15), sigmaX: 0)
13     elif filter_type == 'edge':
14         result[index] = cv2.Canny(image_section, 100, 200)
15     elif filter_type == 'grayscale':
16         result[index] = cv2.cvtColor(image_section, cv2.COLOR_BGR2GRAY)
17     elif filter_type == 'sharpen':
18         kernel = np.array([[0, -1, 0], [-1, 5, -1], [0, -1, 0]]) # Шарпен фильтрі
19         result[index] = cv2.filter2D(image_section, -1, kernel)
20
```

Сурет 2. Фильтрге қолданылған функциялар

Image_section — сүзгі пайдаланатын кескіннің белгілі бір бөлігін білдіретін массив. Ал Filter_type — таңдалатын сүзгінің түрі, яғни функцияны шақыру барысында беріледі (мысалы: blur, edge, grayscale, sharpen). Алынған нәтиже Result массивінде сақталады. Функция орындалып біткен соң, өңделген кескін фрагменттері дәл осы массивте жиналады. Сонымен қатар, нәтижелер тізімінде әрбір кескін бөлігінің сақталу орнын анықтайтын index көрсеткіші болады [10].

Гаусс сүзгісі (Gaussian Blur): Кескінді бұлыңғырлау әдісі оның анықтығын төмендетіп, беткі құрылымын тегістейді. Бұл тәсіл көбінесе шу деңгейін азайту мақсатында қолданылады. Сондай-ақ, кескіндегі біркелкі аймақтарды анықтауға мүмкіндік береді. Мұндай әдіс қажет емес визуалды элементтерден арылу үшін тиімді шешім бола алады. Бағдарламада қолданылатын cv2.GaussianBlur функциясы кескінге Гаусс сүзгісін қолдану арқылы бұлыңғырлық әсерін береді. Бұл жерде ядро өлшемі (15,15) болып таңдалып, стандартты ауытқу мәні 0-ге тең етіп орнатылады.

Сұр ренктер, Грейскейл (Grayscale): Сұр шкалаға ауыстыру әдісі – түрлі-түсті кескінді кара-ақ форматқа түрлендіру арқылы өңдеуді оңайлататын тәсіл. Келесі әдіс кескіннің тек бір арнасы – жарықтылық компоненті бойынша жұмыс істейді. Қажетсіз түсті мәліметтерді алып тастау арқылы өңдеуді жылдам әрі тиімді етуге мүмкіндік береді. Әрбір әрекет cv2.cvtColor функциясын қолдану арқылы жүзеге асады, ал cv2.COLOR_BGR2GRAY – түрлендірудің арнайы коды ретінде қолданылады.

Шарпен сүзгісі (Sharpen): Айқындау сүзгісі кескіннің анықтығын күшейтіп, оның құрылымдық ерекшеліктерін айқын етеді. Объектінің шекараларын бөлектеу үшін маңызды. Сондай-ақ, ұсақ бөлшектерді нақты бейнелеуге мүмкіндік береді. Осы тәсіл кескіннің анықтығын жоғарылату үшін арнайы құрылған конволюциялық ядроны қолданады. Айқындау процесі cv2.filter2D функциясы арқылы жүзеге асады. Мұндағы kernel – контрастты арттыруға арналған арнайы сүзгі өзегі [11].

Функция фильтр түріне қарай сәйкес алгоритмді қолдану арқылы кескіннің белгілі бір аймағын өңдейді. Ол қатар жүргізілетін ағындар арқылы орындалады. Кескін төрт бөлікке бөлінеді және әрқайсысына белгілі бір сүзгі түрі қолданылады. Өңдеу нәтижелері кейін біріктіру үшін ортақ нәтижелер массивінде көрсетіледі.

Apply_filter командасы бейнені өңдеудің логикасын іске асырып, сүзгілер арқылы кескіннің бөліктерін өңдеуге мүмкіндік береді. Параллель өңдеудің маңызды элементі болып табылады.

Жиектерді анықтау, сұр реңк және қаріп сүзгілері сияқты тәсілдер кескін деректерін әртүрлі мақсатта өңдеуге мүмкіндік береді. Әр әдіс өзіндік ерекшеліктерге ие және нақты мәселелерді шешуге бағытталған. Мысалы, жиектерді анықтау нысандардың пішінін белгілеуге көмектеседі, сұр реңк қарапайым өңдеу мен деректерді қысқарту үшін, ал антиалиазинг кескіннің анықтығын жақсартып, бөлшектерді айқындауға мүмкіндік береді.

Көрсетілген функция қатар жүргізілетін өңдеу арқылы бір уақытта бірнеше сүзгіні кескіннің бөліктеріне орындауға мүмкіндік береді. Бұл көп ағынды тәсіл кескінді өңдеу жылдамдығын арттырып, өнімділікті жақсарттады. Осылайша, үлкен кескіндерді өңдеу уақыты қысқарып, жүйенің есептеу ресурстары тиімді пайдаланылады.

Параллель өңдеудің мүмкіндіктері мен сүзгілерді қолдану бейнебақылау, медициналық бейнелеу және уақыттағы кескіндерді өңдеу салаларында аса маңызды болып табылады. Бұл тәсілдер кескіндерді өңдеу және компьютерлік көру саласындағы көптеген күрделі мәселелерді шешуге мүмкіндік береді.[12]

Параллель алгоритмді есептеу және таңдау. Параллель алгоритмдер кескінді өңдеудің тиімділігін арттыру үшін көп ядролы функцияларды қолдануға мүмкіндік береді, сонымен қатар көп ағынды жүйелерді пайдалануды қамтамасыз етеді. Суреттер үлкен көлемдегі деректерден тұрады, бұл олардың әрбір пикселін өңдеу кезінде уақыттың көп жұмсалыуына себеп болады. Егер кескін өте үлкен болса, бұл амал көп уақыт алуы мүмкін. Параллель алгоритмдер кескінді бірнеше бөлікке бөліп, әр бөлікпен бір уақытта жұмыс істей отырып, өңдеу уақытын айтарлықтай қысқартады.

Параллель алгоритмдер кескінді өңдеуде өте тиімді құрал болып табылады, себебі олар өңдеу уақытын айтарлықтай қысқартады. Мұндай алгоритмдер бір уақытта бірнеше есептеулер жүргізуге мүмкіндік береді, бұл үлкен кескіндерді өңдеу кезінде өте маңызды. Параллель алгоритмдер көп ядролы процессорларда және графикалық өңдеу блоктарында (GPU) қолданылған кезде, әрбір ағын немесе процессор ядросы кескіннің бір бөлігін өңдейді, бұл өңдеу жылдамдығын арттырады.

Параллель алгоритмді жүзеге асыру үшін бірнеше негізгі қадамдар орындалады. Алдымен, кескінді бірнеше бөлікке бөлу қажет. Мысалы, кескінді төрт бөлікке бөліп қарастырайық: жоғарғы сол жақ, жоғарғы оң жақ, төменгі сол жақ, төменгі оң жақ. Осылайша, әрбір бөлік үшін өңдеу процессі дербес жүргізіледі.

Әрбір ағын өз бөлігімен жеке жұмыс істейді. Мысалы, бір ағын суреттің бұлыңғырлау сүзгісін қолданса, екінші ағын жиекті анықтау әдісін пайдалана алады, ал үшінші ағын сұр реңкке ауыстыру операциясын орындайды. Осы тәсіл әр ағынның бір уақытта жеке тапсырмасын орындауына мүмкіндік береді, нәтижесінде бүкіл кескіннің өңдеу уақыты айтарлықтай қысқарады.

Әрбір ағын өз жұмысын аяқтаған соң, олардың нәтижелері біріктіріледі. Әр қадамның маңызы зор, себебі әрбір бөлік өңделіп болғаннан кейін, алынған деректерді қайтадан біріктіріп, түпкілікті кескінді алу қажет болады. Нәтижесінде, барлық бөлік өңделіп болғаннан кейін, кескін толық әрі өңделген түрде дайын болады.

Мысалы, егер біз үлкен кескінді төрт бөлікке бөліп, әр бөлікті жеке ағындарда өндесек, өңдеу уақытын төрт есе қысқартуға болады. Нақты уақыттағы бейнемазмұнды өңдеу кезінде өте маңызды, өйткені уақытты үнемдеуге мүмкіндік береді.

Параллель алгоритмдер бірнеше сүзгіні бір уақытта қолдануға мүмкіндік береді. Әрбір ағын өз бөлігінде әртүрлі әдістерді қолдана алады, мысалы, бұлыңғырлау, жиекті анықтау немесе сұр реңк сүзгілерін қолдану. Осы әдіс өңдеу кезінде әртүрлі тапсырмаларды параллель орындауға мүмкіндік беріп, жалпы өңдеу уақытын қысқартуға ықпал етеді [13].

Сонымен, қатар жүргізілетін алгоритмдер кескінді өңдеу жылдамдығын арттыруда өте тиімді құрал болып табылады, себебі олар көп ресурстарды параллельді түрде қолдануға мүмкіндік береді.

Параллель кескінді өңдеу үлкен кескіндермен жұмыс істегенде немесе күрделі кескіндерді өңдеу кезінде өңдеу уақытын қысқартуға мүмкіндік береді. Келесі әдіс кескінді бірнеше бөлікке бөліп, әрбір бөлікке қатар жүргізілетін өңдеу сүзгілерін қолдануға мүмкіндік береді, осылайша есептеулерді бір уақытта орындап, жалпы өңдеу уақытын едәуір қысқартуға мүмкіндік береді (3-сурет).


```
22 def process_image(image_path, filter_type):
23     """Суретті бөліп, параллель өңдеу жасау."""
24     image = cv2.imread(image_path)
25     height, width = image.shape[:2]
26     # Суретті төрт бөлікке бөлу
27     sections = [
28         image[0:height // 2, 0:width // 2], # сол жақ жоғары бөлік
29         image[0:height // 2, width // 2:], # оң жақ жоғары бөлік
30         image[height // 2, 0:width // 2], # сол жақ төменгі бөлік
31         image[height // 2, width // 2:] # оң жақ төменгі бөлік
32     ]
33     # Нәтижені сақтау үшін орын
34     results = [None] * 4
35     threads = []
36     # Әрбір бөлікке фильтр қолдану үшін ағымдар құру
37     for i in range(4):
38         t = threading.Thread(target=apply_filter, args=(sections[i], filter_type, results, i))
39         threads.append(t)
40         t.start()
41     # Ағымдардың аяқталуын күту
42     for t in threads:
43         t.join()
44     # Бөлінген бөліктерді қайта жинақтау
45     if filter_type == 'grayscale': # Грейскейн үшін нәтижелерді бір каналға келтіру
46         final_image = np.vstack((np.hstack((results[0], results[1])), np.hstack((results[2], results[3]))))
47         final_image = cv2.cvtColor(final_image, cv2.COLOR_GRAY2BGR) # Грейскейнді BGR форматына келтіру
48     else:
49         final_image = np.vstack((np.hstack((results[0], results[1])), np.hstack((results[2], results[3]))))
50     return final_image
```

Сурет 3. Суретті бөліп, Қатар жүргізілетін өңдеу жасау

Кескіндерді параллель өңдеу әдісін қолдану бүгінгі таңда көптеген маңызды салаларда тиімді шешімдер ұсынуда. Келесі әдістің негізінде үлкен кескіндер мен бейнемазмұнды өңдеу процесін жеделдету, жүйелік ресурстарды тиімді пайдалану жатыр. Бағдарламаның негізгі бөліктерін талдай отырып, әр қадамның маңыздылығын түсіну маңызды.

Кескін `cv2.imread` функциясы арқылы жүктеледі. Бұл функция кескінді ашып, оны өңдеуге дайындайды. Мұнда кескіннің өлшемін өзгерту немесе оны басқа форматта сақтау сияқты әрекеттер орындалуы мүмкін. Одан кейін, кескін төрт бөлікке бөлінеді: жоғарғы сол жақ, жоғарғы оң жақ, төменгі сол жақ және төменгі оң жақ. Бұл бөлу кескінді өңдеудің параллельді түрде орындалуына мүмкіндік береді, себебі әр бөлік жеке өңделеді.

Параллель өңдеу үшін әрбір бөлікке төрт ағын (жіп) бөлінеді. Әрбір ағын кескіннің тек бір бөлігін өңдейді. Ағындар `application_filter` деп аталады және әр ағын өз бөлігіне белгілі бір сүзгіні қолданады (мысалы, бұлыңғырлау, жиекті анықтау немесе сұр реңк сүзгісі). Әр ағын өзінің жұмысын аяқтағаннан кейін, нәтижелері тізімге жазылады. Барлық ағындар өз жұмысын аяқтаған соң, нәтижелер қайта біріктіріледі.

Ағынның жұмысын аяқтауынан кейін, нәтижелер қайта біріктіріледі, және нәтижесінде толық өңделген кескін алынады. Қолданылған сүзгі сұр реңк болса, нәтиже бір арнаға (monochrome) дейін қысқарады. Ал егер басқа сүзгілер қолданылса, кескіннің бастапқы BGR (көгілдір, жасыл, қызыл) форматында сақталуы мүмкін. Бұл кадам әртүрлі сүзгілер мен өңдеу әдістерін қолданудың маңыздылығын көрсетеді.

Параллель өңдеу әдісі жүйелік ресурстарды тиімді пайдаланып, өңдеу уақытын қысқартады. Әрбір ағын өзінің бөлігімен жұмыс істеген кезде процессор мен жады ресурстары оптималды түрде қолданылады. Үлкен көлемдегі деректермен жұмыс істегенде өте маңызды, себебі жүйенің жалпы өнімділігі мен жылдамдығы айтарлықтай артады[13].

Бейнемазмұнды өңдеу кезінде, әсіресе бейнебақылау жүйелерінде немесе медициналық кескіндерде, өңдеу уақытын қысқарту өте маңызды. Параллель өңдеу әдісі бұл мәселелерді шешуге үлкен көмегін тигізеді, себебі кескінді өңдеу бірнеше бөлікке бөлініп, әр бөлік жеке ағындармен өңделеді.

Бұл әдіс қазіргі уақытта көптеген жаңа технологияларда қолданылуда. Мысалы, автономды көліктер, кеңістік деректерін өңдеу және жылдам бейне өңдеу сияқты нақты уақыттағы жүйелерде қатар жүргізілетін алгоритмдер жоғары өнімділік көрсетеді. Бұл әдіс өңдеу процесінің күрделілігін жеңілдетеді, үлкен көлемдегі мәліметтерді өңдеудің жылдамдығын арттырады, және ресурстарды тиімді пайдалануға жол ашады.

Талдау мен нәтижелер

Python тілінде Tkinter қосымшасын пайдалану ол алгоритмді жүзеге асыруды қамтамасыз етеді. Функцияны пайдаланушыға суретті таңдауға және таңдалған сүзгіні қолдануға мүмкіндік береді. Экранда Tkinter қосымшасы арқылы өңделген сурет көрсетіледі. Кескінді өңдеу процесін пайдаланушыға ыңғайлы етеді (4-сурет).

```
57 def open_image():
58     """Суретті таңдап, фильтр қолдану."""
59     file_path = filedialog.askopenfilename()
60     if file_path:
61         # Фильтр түрін алу
62         filter_type = filter_var.get()
63         processed_image = process_image(file_path, filter_type)
64
65         # OpenCV суретінің форматынан PIL форматына ауыстыру
66         processed_image_rgb = cv2.cvtColor(processed_image, cv2.COLOR_BGR2RGB)
67         img = Image.fromarray(processed_image_rgb)
68         img_tk = ImageTk.PhotoImage(img)
69
70         # Суретті экранда көрсету
71         display_label.config(image=img_tk)
72         display_label.image = img_tk
73
```

Сурет 4. Таңдалған суретке фильтр қолдану

Бұл бағдарлама пайдаланушыға сурет таңдау мүмкіндігін ұсыну және таңдалған суретті белгілі бір сүзгілермен өңдеп, нәтижесін. Қадамдарды түсіну үшін әр бөлікті бөлшектеп қарастырайық.

1. Суретті таңдау (File Dialog)

Алғашқы қадам — пайдаланушыдан сурет таңдау үшін **FileDialog.askopenfilename()** функциясын қолдану. Бұл функция пайдаланушыға файл таңдау терезесін ашады. Пайдаланушы файлды таңдап, оның жолын таңдағаннан кейін, бұл жолды қайтарады.

python

Көшіру/өңдеу

file_path = filedialog.askopenfilename() # Пайдаланушыдан файлды таңдау

Бұл код суреттің файл жолын қайтарады. Пайдаланушы өз файлын таңдағаннан кейін, келесі қадамға өтеміз.

2. Сүзгі түрін алу (filter_var.get())

Сүзгі түрін таңдау үшін **Tkinter** интерфейсінде пайдаланушыға мүмкін таңдаулар ұсынылады. Бұл таңдаулар **filter_var.get()** арқылы алынады. Мұнда **filter_var** — Tkinter жүйесінде анықталған **StringVar** типіндегі айнымалы, ол пайдаланушының таңдаған сүзгісін сақтайды [13].

3. Суретті өңдеу (process_image функциясы)

Сурет **process_image** функциясы арқылы өңделеді. Осы функцияға екі параметр жіберіледі: **file_path** (пайдаланушы таңдаған файл жолы) және **filter_type** (пайдаланушы таңдап алған сүзгі түрі). Бұл функция көрсетілген сүзгімен кескінді өңдейді және қайтарады.

python

Көшіру/өңдеу

processed_image = process_image(file_path, filter_type) # Суретті өңдеу

Бұл қадамда сурет таңдалып, өңдеу үшін фильтр түрі беріледі. Одан кейінгі қадамда кескін өңделіп, дайын болады.

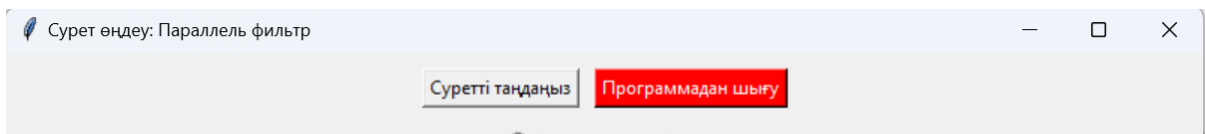
4. BGR-дан RGB-ге түрлендіру

OpenCV кітапханасы кескіндерді **BGR** пішімінде оқиды, бірақ **Tkinter** интерфейсінде суреттерді көрсету үшін **RGB** пішімінде болуы қажет.

```
75 # Tkinter интерфейсі
76 root = tk.Tk()
77 root.title("Сурет өңдеу: Параллель фильтр")
78
79 # Батырмаларды орналастыру үшін жаңа Frame жасаймыз
80 button_frame = tk.Frame(root)
81 button_frame.pack(pady=10)
82
83 # Сурет таңдау батырмасы
84 select_button = Button(button_frame, text="Суретті таңдаңыз", command=open_image)
85 select_button.grid(row=0, column=0, padx=5)
86
87 # Программадан шығу батырмасы
88 exit_button = Button(button_frame, text="Программадан шығу", command=root.quit, bg="red", fg="white")
89 exit_button.grid(row=0, column=1, padx=5)
90
```

Сурет 5. Tkinter графикалық интерфейсін құру

Tkinter терезесін құру үшін алдымен root деп аталатын негізгі терезе жасалады. Бұл код екі түймеден тұратын Tkinter терезесін ашады: біреуі суретті таңдап, өңдеуді бастауға мүмкіндік береді, ал екіншісі бағдарламадан шығуды қамтамасыз етеді[14]. Бұл интерфейс пайдаланушыға суретпен жұмыс істеу мүмкіндігін және бағдарламаны жабуды ұсынады (6-сурет).



Сурет 6. Tkinter терезесінің нәтижесі

Пайдаланушыға суретті таңдауға және оған әртүрлі сүзгілерді қолдануға мүмкіндік береді. Белгі кескіндерді көрсетуге мүмкіндік береді. Қосымша қарапайым және пайдаланушыға ыңғайлы (7-сурет).

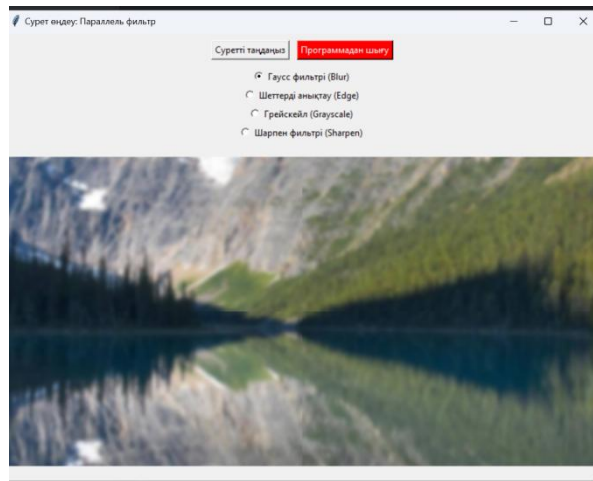
```
91 # Филтрге таңдау
92 filter_var = tk.StringVar(value="blur") # Әдепкі фильтр - blur
93 blur_radio = tk.Radiobutton(root, text="Ғайсс Филтрге (Blur)", variable=filter_var, value="blur")
94 edge_radio = tk.Radiobutton(root, text="Шеттерді анықтау (Edge)", variable=filter_var, value="edge")
95 grayscale_radio = tk.Radiobutton(root, text="Ғрейскейл (Grayscale)", variable=filter_var, value="grayscale")
96 sharpen_radio = tk.Radiobutton(root, text="Шарпен Филтрге (Sharpen)", variable=filter_var, value="sharpen")
97
98 blur_radio.pack()
99 edge_radio.pack()
100 grayscale_radio.pack()
101 sharpen_radio.pack()
102
103 # Суретті көрсету
104 display_label = Label(root)
105 display_label.pack(pady=20)
106
107 # Tkinter қолданбасын іске қосу
108 root.geometry("800x600")
109 root.mainloop()
110
```

Сурет 7. Tkinter интерфейсінде филтрге таңдау және суретті көрсету

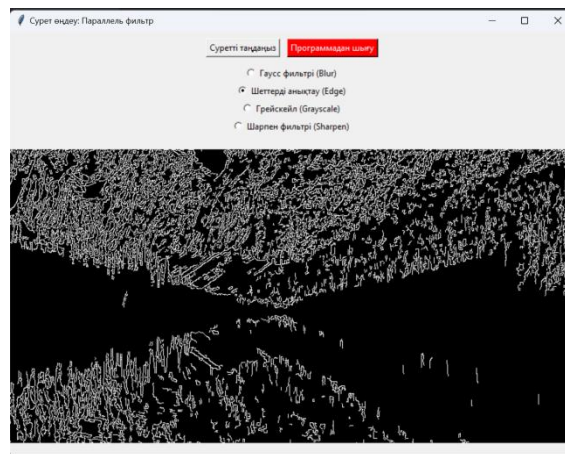
Бұл бағдарлама үзіндісі Tkinter интерфейсінде пайдаланушыларға сүзгі таңдау мүмкіндігін береді, сонымен қатар кескінді көрсетуге де мүмкіндік береді. Әрбір бөлікті толығырақ қарастырып өтейік: filter_var - бұл Tkinter-дің StringVar типіндегі айнымалысы.

Пайдаланушы түймелерді басқан кезде кескін таңдау және сүзгілерді қолдану сияқты әрекеттер орындалады [15].

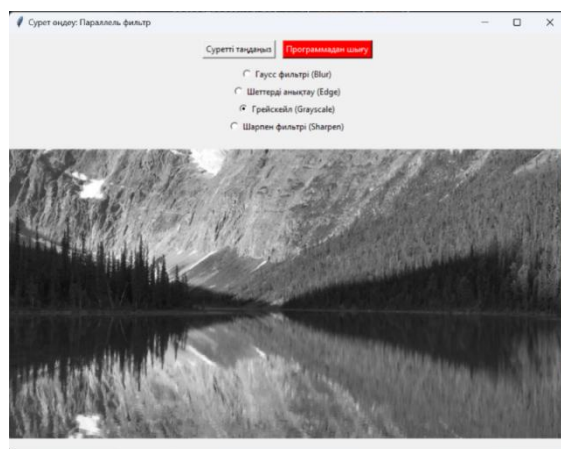
Осы код арқылы пайдаланушыларға сүзгіні таңдап, оны суретке қолдануға мүмкіндік беріледі, ал нәтижелер экранда көрсетіледі (8,9,10,11-сурет).



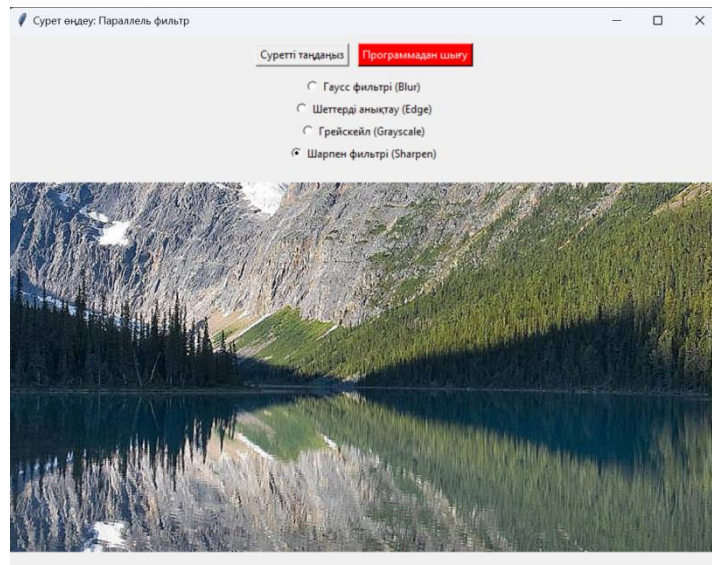
Сурет 8. Гаусс фильтр нәтижесі



Сурет 9. Шеттерді анықтау фильтр нәтижесі



Сурет 10. Грейскейл фильтр нәтижесі



Сурет 11. Шарпен фильтри нәтижесі

Қорытынды

Параллель алгоритмдерді қолдану кескінді өңдеу процестерінің жылдамдығын айтарлықтай арттырады. Осы әдіс жүйелік ресурстарды тиімді пайдалануға және үлкен көлемдегі деректермен жұмыс істеуде өңдеу уақытын қысқартуға мүмкіндік береді. Әдетте, кескінді өңдеу үлкен көлемдегі деректермен жұмыс істегенде, ол көп уақыт пен ресурстарды қажет етеді. Алайда, қатар жүргізілетін өңдеу әдісінде бұл процесс бірнеше кішігірім тапсырмаларға бөлінеді, және әр тапсырма бір уақытта орындалады. Әрбір тапсырма параллельді түрде орындалғандықтан, өңдеу уақыты айтарлықтай қысқарады.

Параллель алгоритмдерді қолдана отырып, кескіннің әр бөлігіне бір уақытта әртүрлі сүзгілерді қолдануға болады. Осы әдіс жүйенің жұмысын айтарлықтай жақсартады, себебі ол бір уақытта бірнеше тапсырмаларды орындауға мүмкіндік береді. Өңдеу уақытын қысқартуға және жүйе ресурстарын тиімді пайдалануға ықпал етеді.

Параллель алгоритмдер көп ядролы процессорлар мен графикалық процессорларды (GPU) тиімді пайдалануға мүмкіндік береді. Мұндай ресурстарды қолдану, әсіресе үлкен көлемдегі кескіндерді өңдеу кезінде, өнімділікті айтарлықтай арттырады. Бірнеше ағынды және графикалық өңдеу құралдарын пайдалану, жүйенің жалпы өнімділігін арттырып, кескінді өңдеу уақытын едәуір қысқартады.

Параллель алгоритмдер кескіндерді өңдеудің тиімділігін айтарлықтай арттырады, әсіресе нақты уақыттағы бейнеконтентпен жұмыс істегенде. Сонымен қатар, жүйе ресурстарының тиімді пайдаланылуы, қуат тұтынуды азайтуға және ұзақ мерзімді тиімділікке ықпал етеді.

Параллель алгоритмдер қазіргі уақытта түрлі салада қолданыс табуда, оның ішінде автономды көліктер, кеңістік деректерін өңдеу, нақты уақыттағы бейнеконтентті өңдеу және медициналық кескіндер сияқты салаларда. Технологиялардың дамуы мен қолданылуы үлкен деректермен жұмыс істеу тиімділігін арттырып, өңдеу процесінің күрделілігін жеңілдетеді.

Параллель алгоритмдер кескіндерді өңдеу уақытын едәуір қысқартып, жүйенің өнімділігін арттырады. Әр түрлі сүзгілерді қолдану кезінде параллельді өңдеу әдісінің артықшылықтары айқын көрінеді. Осы әдіс жүйелік ресурстарды тиімді пайдаланып, үлкен деректермен жұмыс істегенде өңдеу жылдамдығын арттырады. Сонымен қатар, бұл әдіс ресурстарды оңтайлы пайдалану және қуат үнемдеу сияқты қосымша артықшылықтарға ие. Параллель алгоритмдер бүгінгі таңда бейнеконтентті өңдеудің маңызды құралы болып табылады және болашақта бұл технологияның дамуы жаңа мүмкіндіктерге жол ашады.

Әдебиеттер тізімі

1. Grama, A., Gupta, A., Karypis, G., & Kumar, V. (2003). *Introduction to parallel computing*. Addison Wesley.
2. Двойнишиников, С. В., Меледин, В. Г., & Павлов, В. А. (2017). Высокоскоростная обработка фазовых изображений с использованием параллельных вычислений. *Автометрия*, 53(2), 56–62.
3. Клебан, Е. Г. (2020). *Параллельный алгоритм обработки изображений и видео*.
4. Dey, S. (2020). *Python image processing cookbook*. Packt Publishing.
5. Szeliski, R. (2020). *Computer vision: Algorithms and applications*. Springer.
6. Смайылов, Ж. (2017). Мәліметтерді параллель өңдеудегі тиімді әдістер. *Ақпараттық технологиялар журналы*, (1).
7. Базарбаев, С. (2018). Кескіндерді параллель өңдеу: заманауи тәсілдер. *Қазіргі заманғы ғылым және техника*, (4).
8. Әбдірахманов, Н. (2021). Параллель алгоритмдердің өнімділігі: сурет талдаудағы тәжірибелер. *Ғылым және инновациялар журналы*, (2).
9. Жарылқасын, Е. (2022). Суреттерді өңдеудегі параллель сұрыптау әдістері. *Ақпараттық жүйелер журналы*, (1).
10. Нұрпейісова, А. А. (2021). *Инновациялық өнімді құру процесін оңтайландырудың математикалық моделін дайындау*.
11. Елешова, Ш. Д., және т.б. (2021). Мобильді құрылғыға арналған электронды оқулықтарды әзірлеу ортасын талдау. *Қ. Жұбанов атындағы Ақтөбе өңірлік университеті еңбектері*, 38.
12. Shausheva, A. G., et al. (2022). Problems of speech recognition based on artificial neural networks in the proctoring system. *Physico-Mathematical Series*, 148.
13. Мырзамуратова, А. А., & Жанабайқызы, О. Г. (2024). Современные вызовы и тенденции в защите информации: стратегии и технологии. *Еңбектер жинағы / Сборник трудов / Collection of works*, 292.
14. Шаушенова, А., және т.б. (2022). Прокторинг жүйесінде жасанды нейрондық желілерге негізделген сөйлеуді тану мәселелері. *Известия НАН РК. Серия физико-математическая*, 4, 146–158.
15. Бауыржанұлы, Б., және т.б. (2021). Algorithm for biometric identification of a person in open systems. *Интернаука*, (18-4), 61–66.

References

1. Grama, A., Gupta, A., Karypis, G., & Kumar, V. (2003). *Introduction to parallel computing*. Addison Wesley.
2. Dvoynishnikov, S. V., Meledin, V. G., & Pavlov, V. A. (2017). Vysokoskorostnaya obrabotka fazovykh izobrazhenii s ispol'zovaniem parallel'nykh vychislenii. *Avtometriya*, 53(2), 56–62.
3. Kleban, E. G. (2020). *Parallel'nyi algoritm obrabotki izobrazhenii i video*.
4. Dey, S. (2020). *Python image processing cookbook*. Packt Publishing.
5. Szeliski, R. (2020). *Computer vision: Algorithms and applications*. Springer.
6. Smayilov, Zh. (2017). Tiimdi adister malimetterdi parallel ondeude. *Aqparattyq tehnologiyalar jurnaly*, (1).
7. Bazarbaev, S. (2018). Keskinderdі parallel ondeu: zamañy tasilder. *Qazirgi zamangy nylym jane tekhnika*, (4).
8. Abdirakhmanov, N. (2021). Parallel algoritmderdin onimdiligi: suret taldaydagı tajiribeler. *Gylym jane innovatsiyalar jurnaly*, (2).
9. Jarylkasyn, E. (2022). Syretterdi ondeydegi parallel surypay adisteri. *Aqparattyq juierler jurnaly*, (1).
10. Nurpeisova, A. A. (2021). *Innovatsiyalyq onimdi quru prosesin ontailylandyrudyn matematikalyq modelin дайындау*.
11. Elshova, Sh. D., et al. (2021). Mobil'di qurylgyga arналған elektrondy oqulyqtardy azirleu ortasyn taldaу. *Q. Zhubanov atyndagı Aqtobe onirlik yniversiteti enbekteri*, 38.
12. Shausheva, A. G., et al. (2022). Problems of speech recognition based on artificial neural networks in the proctoring system. *Physico-Mathematical Series*, 148.
13. Myrzamuratova, A. A., & Zhanabaikyzы, O. G. (2024). Sovremennye vyzovy i tendentsii v zashchite informatsii: strategii i tekhnologii. *Collection of Works*, 292.
14. Shausheva, A., et al. (2022). Proctoring juiesinde jasandy neirondyq jelilerge negizdelgen sóleydi tanu máselesi. *Izvestiya NAN RK. Seriya fiziko-matematicheskaya*, 4, 146–158.
15. Bauyrzhanuly, B., et al. (2021). Algorithm for biometric identification of a person in open systems. *Internauka*, (18-4), 61–66.

Авторлар туралы мәліметтер

№	Аты-жөні, ғылыми дәрежесі, жұмыс немесе оқу орны, қала, ел, автордың e-mail мекенжайы, ұялы телефон нөмірі
1	Баялы А.Т. - аға оқытушы, Қожа Ахмет Ясауи атындағы Халықаралық қазақ-түрік университеті, Түркістан қ., Қазақстан, e-mail: azimkhan.bayaly@ayu.edu.kz
	Bayaly A.T. - Senior Lecturer, Akhmet Yassawi International Kazakh-Turkish University, Kazakhstan, Turkistan, e-mail: azimkhan.bayaly@ayu.edu.kz
	Баялы А.Т. - старший преподаватель, Международный казахско-турецкий университет имени Ходжи Ахмеда Ясави, г. Туркестан, Казахстан, e-mail: azimkhan.bayaly@ayu.edu.kz
2	Жунисов Н.М. - PhD, аға оқытушы, Қожа Ахмет Ясауи атындағы халықаралық қазақ-түрік университеті, Түркістан қ., Қазақстан, e-mail: nurseit.zhunissov@ayu.edu.kz
	Zhunisso N.M. - PhD, International Kazakh-Turkish University named after Khoja Akhmet Yasawi, Turkistan, Kazakhstan, e-mail: nurseit.zhunissov@ayu.edu.kz
	Жунисов Н.М. - PhD, старший преподаватель, Международный казахско-турецкий университет имени Ходжи Ахмеда Ясави, г. Туркестан, Казахстан, e-mail: nurseit.zhunissov@ayu.edu.kz
3	Назарбек Т.С. - магистр оқытушы, Қожа Ахмет Ясауи атындағы Халықаралық қазақ-түрік университеті, Түркістан қ., Қазақстан, e-mail: Tolkin.Nazarbek@ayu.edu.kz
	Nazarbek T.S. - master lecturer, Akhmet Yassawi International Kazakh-Turkish University, Kazakhstan, Turkistan, e-mail: Tolkin.Nazarbek@ayu.edu.kz
	Назарбек Т.С. - магистр преподаватель, Международный казахско-турецкий университет имени Ходжи Ахмеда Ясави, г. Туркестан, Казахстан, e-mail: Tolkin.Nazarbek@ayu.edu.kz
4	Давлетова В. М. - магистр оқытушы, Қожа Ахмет Ясауи атындағы Халықаралық қазақ-түрік университеті, Түркістан қ., Қазақстан, e-mail: venera.davletova@ayu.edu.kz
	Davletova V.M. - master lecturer, Akhmet Yassawi International Kazakh-Turkish University, Kazakhstan, Turkistan, e-mail: venera.davletova@ayu.edu.kz
	Давлетова В.М. - магистр преподаватель, Международный казахско-турецкий университет имени Ходжи Ахмеда Ясави, г. Туркестан, Казахстан, e-mail: venera.davletova@ayu.edu.kz