

ҚАЗАҚСТАН РЕСПУБЛИКАСЫНЫҢ БІЛІМ ЖӘНЕ ҒЫЛЫМ МИНИСТРЛІГІ

«Қ. И. Сәтбаев атындағы Қазақ ұлттық техникалық зерттеу университеті»  
коммерциялық емес акционерлік қоғамы



Автоматтандыру және ақпараттық технологиялар институты  
«Роботтық техника және автоматиканың техникалық құралдары» кафедрасы

Бақыт Арман Нұржанұлы

«Машиналық оқыту арқылы піскен қызанақты тану жүйесі»

дипломдық жобаға  
ТҮСІНДІРМЕ ЖАЗБА

6B07113 - Робототехника и мехатроника

ҚАЗАҚСТАН РЕСПУБЛИКАСЫНЫҢ БІЛІМ ЖӘНЕ ҒЫЛЫМ МИНИСТРЛІГІ

«Қ. И. Сәтбаев атындағы Қазақ ұлттық техникалық зерттеу университеті»  
коммерциялық емес акционерлік қоғамы



SATBAYEV  
UNIVERSITY

Автоматтандыру және ақпараттық технологиялар институты  
«Роботтық техника және автоматиканың техникалық құралдары» кафедрасы

**ҚОРҒАУҒА ЖІБЕРІЛЕДІ**  
РТжАТҚ кафедра меңгерушісі  
техника ғылымының кандидаты,  
профессор

Қ. Ә. Өжікенов

« 6 » 2024 ж.



**дипломдық жобаға  
ТҮСІНДІРМЕ ЖАЗБА**

Тақырыбы: «Машиналық оқыту арқылы піскен қызанақты тану  
жүйесі»

6B07113 - Робототехника и мехатроника

Орындаған

Бақыт А.Н.

Рецензент

Ғылыми жетекшісі

әл-Фараби атындағы ҚазҰУ,

Жасанды интеллект және

Big Data кафедрасының доценті, PhD

Т.Ғ.М., аға оқытушы

Карымсакова Н.Т.

Жамуратова М.

коды

аты-жөні

«06» маусым 2024 ж.

«6» маусым 2024 ж.

«Қ. И. Сәтбаев атындағы Қазақ ұлттық техникалық зерттеу университеті»  
коммерциялық емес акционерлік қоғамы



Автоматтандыру және ақпараттық технологиялар институты

«Робототехника және автоматиканың техникалық құралдары» кафедрасы

6B07113 - Робототехника и мехатроника

**БЕКІТЕМІН**

РТЖАТҚ кафедра меңгерушісі  
техника ғылымының кандидаты,  
профессор

  
К. Ә. Өжікенов  
«\_» \_\_\_\_\_ 2024 ж.



**Дипломдық жобаны орындауға арналған  
ТАПСЫРМА**

Білім алушы Бақыт Арман Нұржанұлы

Тақырыбы: Машиналық оқыту арқылы піскен қызанақты тану жүйесі

Университет ректорының 2024 жылғы «\_» \_\_\_ №\_\_ бұйрығымен  
бекітілген

Аяқталған жұмысты тапсыру мерзімі: «\_» мамыр 2024 ж.

Дипломдық жұмыстың бастапқы деректері: PyCharm, YOLOv8, Python.

Дипломдық жұмыста қарастырылатын мәселелер тізімі:

- а) *Машиналық оқыту арқылы піскен қызанақты тану жүйесі жайлы.*
- б) *Қызанақтың піскенін анықтау үшін нейрондарға қолдану*
- в) *Зерттеудің негізгі нәтижелері, машиналық жүйенің әдістері.*

Графикалық материалдың тізбегі (міндетті сызбаларды дәл көрсете отырып):


Жұмыс презентациясы слайдтарда \_ көрсетілген

Ұсынылатын негізгі әдебиеттер: \_ атаулардан


Дипломдық жұмысты (жобаны) дайындау  
**КЕСТЕСІ**

Бөлімдер атауы, әзрленетін сұрақтар тізбегі	Ғылыми жетекшіге ұсыну мерзімдері	Ескертпелер
Теориялық бөлім		Орындалды
Зерттеу бөлімі		Орындалды
Қорытынды бөлім		Орындалды

Аяқталған дипломдық жұмыс (жоба) үшін, оған қатысты бөлімдердің жұмыстарын (жобасын) көрсетумен, кеңесшілері мен қалып бақылаушының қолдары

Бөлімдердің атауы	Кеңесшілер, тегі, аты, әкесінің аты, (ғылыми дәріжесі, атағы)	Қол қойылған күні	Қол
Қалып бақылаушы	Игембай Е.А. техника ғылымдарының магистрі, оқытушы	06.06.2024	

Ғылыми жетекшісі

  
Жамуратова М.

Білім алушы тапсырманы орындауға алды

  
Бақыт А.Н.

Күні

«\_» \_\_\_\_ 2024 ж.

## АҢДАТПА

Бұл дипломдық жобада Машиналық оқыту әдістерін қолдана отырып, піскен қызанақты тану жүйесін зерттеуге арналған.

Зерттеудің негізгі мақсаты тамақ өнімдерін шығаратын кәсіпорындарда қызанақты сұрыптау сапасын жақсарту арқылы, машиналық оқыту жүйесін пайдалану.

Бірінші тарауда машиналық оқыту арқылы піскен қызанақты тану жүйесі жайлы зерттеу әдістері, шешілетін мәселелер және оның шешімі жайлы сипатталады.

Екінші тарауда жұмыста қызанақтың пісу дәрежесін анықтау үшін кескінді өңдеуге және нейрондық желілерді пайдалануға негізделген алгоритм ұсынылады. Тәжірибелер жүйенің тиімділігін және оны егін жинау мен сұрыптау процесін автоматтандыру үшін тәжірибеде қолдану мүмкіндігін көрсетемін. Олар салыстырылып, жалпы кемшіліктер анықталады.

Соңғы тарауда зерттеудің негізгі нәтижелері машиналық жүйенің әдістері ауылшаруашылық кәсіпорындары үшін пайдалы екендігі дәлелденеді.

## АННОТАЦИЯ

Этот дипломный проект посвящен изучению системы распознавания спелых помидоров с использованием методов машинного обучения.

Основной целью исследования является использование системы машинного обучения путем улучшения качества сортировки томатов на предприятиях пищевой промышленности.

В первой главе описываются методы исследования системы распознавания спелых помидоров с помощью машинного обучения, решаемые проблемы и ее решение.

Во второй главе в работе предлагается алгоритм, основанный на обработке изображений и использовании нейронных сетей для определения степени спелости томатов. Эксперименты показывают эффективность системы и возможность ее применения на практике для автоматизации процесса сбора урожая и сортировки. Их сравнивают и выявляют общие недостатки.

В последней главе основные результаты исследования доказывают, что методы машинной системы полезны для сельскохозяйственных предприятий.

## **ABSTRACT**

This graduation project is devoted to the study of the system of recognition of ripe tomatoes using machine learning methods.

The main goal of the study is to improve the quality of tomato sorting at food enterprises, using a machine learning system.

The first chapter describes the methods of studying the system of recognition of ripe tomatoes by machine learning, the problems to be solved and its solution.

In the second chapter, the work proposes an algorithm based on image processing and the use of neural networks to determine the degree of ripeness of tomatoes. Experiments I show the efficiency of the system and the possibility of its application in practice to automate the process of harvesting and sorting. They are compared and general shortcomings are identified.

In the last chapter, the main results of the study prove that machine system methods are useful for agricultural enterprises.

## МАЗМҰНЫ

Кіріспе	4
1 Зерттеудің негізі және маңызы	5
1.1 Зерттеу негізі	5
1.2 Елде және шетелде зерттеу жағдайы	6
1.3 Шешілетін мәселелер мен оларды шешу жолдары	8
1.3.1 Шешілетін мәселелер	8
1.3.2 Шешім	9
1.4 Деректер жиынын өңдеу	11
1.5 Принциптер мен кодекске кіріспе	14
1.5.1 YOLOv8 алгоритмінің принципі	14
1.5.2 Үлгі құрылысы	15
1.5.3 Зерттеудің коды	19
2 Эксперимент нәтижелері және талдау	22
2.1 Зерттеу графиктері	22
2.2 YOLOv8/v7/v6/v5 тәжірибесін салыстыру	25
2.3 Жүйені жобалау	27
2.4 Жүйелік процесс	29
Қорытынды	
Пайдаланылған әдебиеттер тізімі	
Қосымшалар (бағдарлама коды, қосымша материалдар және деректер)	



## КІРІСПЕ

Тақырыптың өзектілігі. Қазіргі заманғы ауыл шаруашылығы өнімділікті арттыру және өндіріс шығындарын азайту міндетіне тап болады. Өнімділікті арттырудың бір жолы-егін жинау және өңдеу процестерін автоматтандыру. Піскен қызанақты ерте пісетін кезеңде тану егін жинау мен сұрыптау процесін оңтайландыруға, шығындарды азайтуға және еңбек шығындарын азайтуға көмектеседі.

Жұмыстың мақсаты. Егін жинау және сұрыптау процесін автоматтандыру үшін машиналық оқыту әдістерін қолдана отырып, піскен қызанақты тану жүйесін жасаңыз.

Жұмыстың міндеттері:

1. Піскен қызанақты танудың қолданыстағы әдістеріне шолу жасаңыз.
2. Піскен және піспеген қызанақтың суреттерін қамтитын оқу деректер жинағын дайындаңыз.
3. Дайындалған деректер жиынтығында Машиналық оқыту моделін үйрету.
4. Піскен және піспеген қызанақтың жаңа суреттерінде модельді сынау.
5. Піскен қызанақты танудың дәлдігін бағалаңыз және басқа тану әдістерімен салыстырыңыз.

Зерттеу әдістері. Жұмыста Машиналық оқыту әдісі, атап айтқанда нейрондық желілерді қолдана отырып терең оқыту қолданылады. Модельді оқыту үшін піскен және піспеген қызанақтың суреттері, сондай-ақ оқу тиімділігін арттыру үшін деректерді күшейту әдістері қолданылады.

Күтілетін нәтижелер. Өзірленген жүйе піскен қызанақты жоғары дәлдікпен тани алады деп күтілуде, бұл ауыл шаруашылығында егін жинау және сұрыптау процесін жақсартуға көмектеседі. Зерттеу нәтижелерін ауыл шаруашылығында автоматтандырылған жүйелерді дамыту үшін пайдалануға болады.

# 1 Зерттеудің негізі және маңызы

## 1.1 Зерттеу негізі

Машиналық оқытуды қолдана отырып, піскен қызанақты тану - бұл саладағы жетістіктерге байланысты соңғы жылдары пайда болған салыстырмалы түрде жаңа зерттеу саласы.

Ерте зерттеулер (2010 ж.):

2014 жылы Мэриленд университетінің зерттеушілері суреттерді пайдалана отырып, піскен қызанақты тану үшін машиналық оқытудың алғашқы үлгілерінің бірін жасады.

2016 жылы Огайо штатының университетінің зерттеушілері қызанақтарды олардың пісуіне қарай жіктеу үшін конволюционды нейрондық желіні (CNN) ұсынды.

Өнімділікті арттыру (2017 ж.):

2017 жылы Урбана-Шампейндегі Иллинойс университетінің зерттеушілері қиын жарықтандыру жағдайында піскен қызанақтарды танудың жоғары дәлдігіне қол жеткізген CNN моделін әзірледі.

2018 жылы Қытай ауылшаруашылық ғылымдары академиясының зерттеушілері қызанақтың жетілуін бір уақытта жіктеуге және ақауларды анықтауға қабілетті көп тапсырмалы терең оқыту моделін ұсынды.

Соңғы жетістіктер (2019 жылдан қазіргі уақытқа дейін):

2019 жылы Техас A&M университетінің зерттеушілері сурет пен спектрлік деректер негізінде піскен қызанақтарды тани алатын машиналық оқыту моделін әзірледі.

2020 жылы Қытай ауылшаруашылық университетінің зерттеушілері мобильді құрылғыларда нақты уақытта піскен қызанақты тану үшін жеңіл салмақты CNN моделін ұсынды.

Ағымдағы трендтер:

Ағымдағы зерттеулер әртүрлі жарықтандыру, фон және қызанақ сорттары сияқты әртүрлі жағдайларда машиналық оқыту үлгілерінің өнімділігін жақсартуға бағытталған.

Зерттеушілер танудың дәлдігін жақсарту үшін гиперспектрлік кескіндер мен сенсор деректері сияқты деректердің басқа түрлерін пайдалануды да зерттеп жатыр.

Сонымен қатар, піскен және піспеген қызанақтарды автоматты түрде ажырата алатын қызанақ сұрыптау жүйелерін әзірлеу үшін машиналық оқыту жүзеге асырылуда.

Осы саладағы зерттеулер ілгерілеген сайын, томаттың піскенін тану жүйелерін машинада оқыту дәлірек және сенімді бола түсуде, бұл оларды

қызанақ өсірушілер, тамақ өңдеушілер және тұтынушылар үшін құнды құралға айналдырады.

## 1.2 Елде және шетелде зерттеу жағдайы

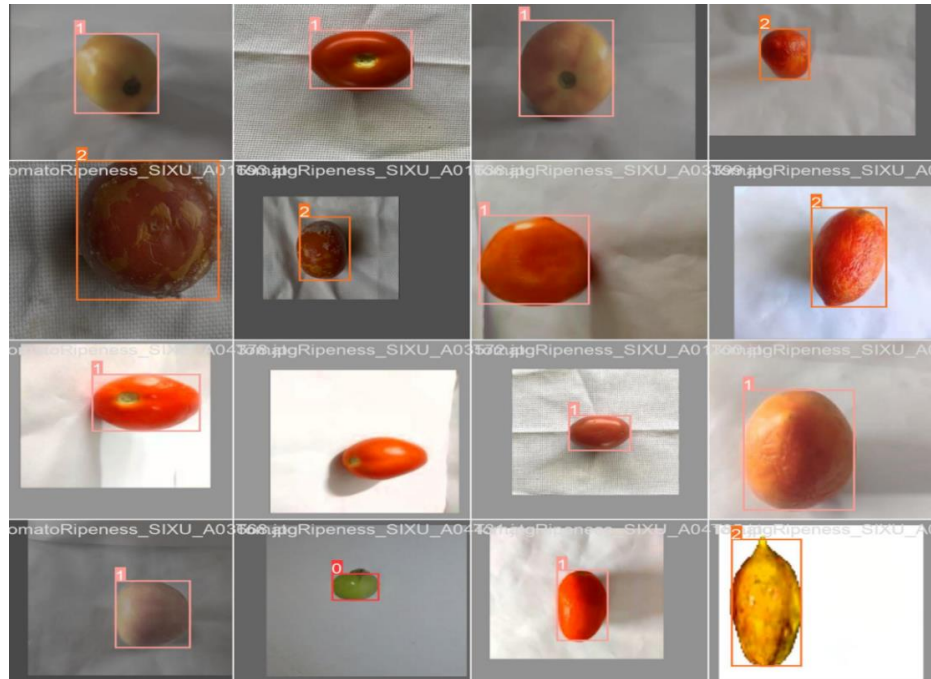
YOLOv8/v7/v6/v5 негізіндегі қызанақтардың пісуін анықтау жүйелерінің ағымдағы зерттеу күйін талқылағанда, біз мақсатты анықтау саласындағы соңғы әзірлемелер мен инновацияларға назар аударуымыз керек. YOLO сериясынан басқа, DETR (Анықтау трансформаторы), ViT (көру трансформаторы), Swin Transformer және терең оқытуға негізделген кейбір инновациялық үлгілер сияқты басқа алгоритмдер де мақсатты анықтау тапсырмаларында айтарлықтай өнімділікті көрсетті.

Ең алдымен, YOLOv8, YOLO сериясының соңғы нұсқасы ретінде, YOLO отбасының жоғары жылдамдықты анықтау артықшылықтарын сақтай отырып, тереңірек және күрделі желі архитектурасын және неғұрлым күрделі функцияларды шығару технологиясын енгізу арқылы анықтау дәлдігін одан әрі жақсартады. YOLOv8 үлгі құрылымы мен оқыту процесін<sup>1</sup> оңтайландыру арқылы бірнеше стандартты деректер жиынында жетекші өнімділікке қол жеткізеді. [1]

DETR нысанды анықтауға Трансформатор архитектурасын енгізеді және жаһандық назар аудару механизмі мен түпкілікті оқыту әдісі арқылы модельдің мақсатты анықтау және орналастыру мүмкіндігін жақсартады. Бұл әдіс күрделі көріністерді өңдеуде және кішігірім нысананы анықтауда<sup>2</sup> жоғары өнімділікті көрсетеді. [2]

ViT Трансформатор архитектурасын кескіндердің блоктық тізбегіне тікелей қолданады және кескінді жіктеу тапсырмаларында өзінің үлкен әлеуетін көрсете отырып, өзіндік назар аудару механизмі арқылы әртүрлі кескін блоктары арасындағы қатынасты түсіреді. Кейіннен бұл идея мақсатты анықтау тапсырмасына кеңейтілді, сәйкес позицияны кодтау және мақсатты анықтау бастарын енгізу арқылы ViT кеңістіктік қатынастарды тиімді өңдей алады және анықтау дәлдігін жақсартады [3] .

Иерархиялық Трансформатор ретінде Swin Transformer тиімді есептеу өнімділігін қамтамасыз етіп қана қоймайды, сонымен қатар өзіне назар аудару ауқымын шектеу және оны біртіндеп кеңейту арқылы модельдің экспрессивті қабілетін жақсартады. Swin Transformer мақсатты анықтау және кескінді сегменттеу сияқты тапсырмаларда тамаша өнімділік көрсетті және икемділігі мен тиімділігінің арқасында зерттеу нүктесіне айналды [4] .



1.1-сурет – PyCharm арқылы суретті өңдеу

Glod-YOLO – шағын мақсатты анықтау өнімділігін жақсарту үшін арнайы жасалған жетілдірілген YOLO үлгісі. Функцияларды шығару желісін жақсарту және жаһандық және жергілікті мүмкіндіктердің оңтайландырылған комбинациясын енгізу арқылы Glod-YOLO өсімдік ауруларын егжей-тегжейлі анықтау үшін пайдалы YOLO сериясын жоғары жылдамдықпен анықтауды сақтай отырып, шағын нысандарды анықтау дәлдігін айтарлықтай жақсартты. және ауылшаруашылық өрісіндегі жәндіктердің зиянкестері әсіресе маңызды. [5]

Томаттың піскендігін анықтауды қолдануда мақсатты анықтаудың осы жетілдірілген алгоритмдерін таңдау және реттеу анықтаудың дәлдігі мен тиімділігін арттыру үшін өте маңызды. Осы алгоритмдердің артықшылықтары мен инновацияларын біріктіре отырып, біз әр түрлі жетілу деңгейлері бар қызанақтарды дәл анықтап, жіктеп қана қоймай, сонымен қатар практикалық ауылшаруашылық өндірісін жетілдіре отырып, әртүрлі орта мен фонға бейімделе алатын жоғары өнімді жетілуді анықтау жүйесін құрастыра алдық. қолданбаның мәні.

## 1.3 Шешілетін мәселелер мен оларды шешу жолдары

### 1.3.1 Шешілетін мәселелер

YOLOv8/v7/v6/v5 негізіндегі қызанақтың пісуін анықтау жүйесін жасау кезінде біз кездестірген қиындықтар негізінен келесі аспектілерге назар аударды:

1. Томаттың пісуін анықтаудың дәлдігі мен жылдамдығы: Қызанақтардың әртүрлі пісетін кезеңдеріндегі сыртқы түрінің өзгерістері түс градиенттері, өлшемдердің шамалы өзгеруі және т.б. сияқты нәзік болады, бұл модельдің тану мүмкіндіктеріне жоғары талаптар қояды. Сонымен қатар, автоматтандырылған сұрыптау желілері сияқты практикалық қолданбалы сценарийлерге бейімделу үшін жүйе нақты уақыт режимінде кескіндерді өңдеуді және талдауды қажет етеді, бұл модельдің дәл ғана емес, сонымен қатар жылдам болуын талап етеді.

2. Қоршаған ортаға бейімделу және үлгіні жалпылау қабілеті: Қызанақтың піскендігін анықтау әртүрлі ортада әртүрлі қиындықтарға тап болуы мүмкін, соның ішінде әртүрлі жарық жағдайлары, фондық кедергі және қызанақ сорттарының әртүрлілігі. Бұл модельдің жалпылау қабілетінің жақсы болуын, қоршаған ортаның әртүрлі жағдайларына бейімделуін және қызанақтың әртүрлі сорттарына жақсы тану әсерін талап етеді.

3. Пайдаланушы интерфейсінің интуитивтілігі мен функционалдығы: Ауыл шаруашылығы қызметкерлері немесе тамақ өңдеу зауытының операторлары жүйені оңай пайдалану үшін интуитивті және қолдануға оңай пайдаланушы интерфейсін әзірлеу қажет. Интерфейс суреттерді, бейнелерді және нақты уақыттағы камераларды енгізуді қолдап қана қоймай, сонымен қатар пайдаланушыларға әртүрлі анықтау қажеттіліктеріне бейімделу үшін әртүрлі модельдер арасында оңай ауысуға мүмкіндік беруі керек.

4. Деректерді өңдеу мүмкіндігі және сақтау тиімділігі: Томаттың піскендігін анықтау жүйесі кескін және бейне деректерінің үлкен көлемін өңдеуді қажет етуі мүмкін екенін ескере отырып, жүйенің деректерді өңдеу мүмкіндігі нақты уақытта анықтауды қамтамасыз ету үшін жеткілікті қуатты болуы керек. Сонымен қатар, сынақ нәтижелері мен тарихи деректерді ұзақ уақыт сақтау үшін тиімді деректерді сақтау механизмі де маңызды.

5. Жүйенің ауқымдылығы және техникалық қызмет көрсету мүмкіндігі: Технологиялар дамып, пайдаланушы қажеттіліктері өзгерген сайын жүйеге жаңа алгоритмдер немесе функциялар енгізу қажет болуы мүмкін. Сондықтан, болашақта жаңа технологиялар үздіксіз интеграциялануы үшін жүйелерді масштабтау мүмкіндігін ескере отырып жобалау қажет. Бұған қоса, жүйенің тұрақты жұмыс істеуі және оңай жаңартылуы мен жаңартылуы үшін оның техникалық қызмет көрсету мүмкіндігі де өте маңызды.

Жоғарыда аталған мәселелерді шешу үшін біз YOLOv8/v7/v6/v5 сияқты терең оқыту үлгілерін үйрету үшін PyTorch негізіндегі техникалық негізді қолданамыз. Бұл модельдер объектіні анықтау саласындағы тиімділігі мен дәлдігі үшін таңдалды. Томаттың піскендігін анықтау сипаттамаларына сүйене

отырып, біз осы тапсырмадағы өнімділігін жақсарту үшін үлгіні тиісті түрде реттедік және оңтайландырдық. Сонымен бірге біз көптеген қызанақ кескіндерін жинап, ұйымдастырдық және модельдің өнімділігін тексеру және оқыту үшін жоғары сапалы деректер жиынтығын жасадық.

Қолданушының өзара әрекеттесуі тұрғысынан біз Streamlit негізінде интуитивті веб-интерфейс әзірледік және CSS әсемдеу арқылы пайдаланушы тәжірибесін жақсарттық. Интерфейс пайдаланушыларға деректерді оңай жүктеп салуға және анықтау нәтижелерін алуға мүмкіндік беретін кескін, бейне және тікелей камера кірісін қолдайды. Сонымен қатар, интерфейс пайдаланушыларға нақты қажеттіліктерге негізделген анықтау үшін ең қолайлы үлгіні таңдауға мүмкіндік беретін әртүрлі үлгі файлдары арасында ауысуды қолдайды.

Бүкіл жүйені әзірлеу барысында біз IDE ретінде PyCharm пайдаланамыз, ол әзірлеу тиімділігін арттырып қана қоймай, кодтың тұрақтылығын қамтамасыз етеді. Осы технологиялар мен стратегиялар сериясы арқылы.

### 1.3.2 Шешім

YOLOv8/v7/v6/v5 негізінде қызанақтың пісуін анықтаудың тиімді және дәл жүйесін құру үшін біз келесі кешенді техникалық шешімдерді қабылдауды жоспарлап отырмыз:

#### 1. Терең оқыту үлгілерін таңдау және оңтайландыру

Модельдік архитектура: Біз YOLOv8 жүйесін негізгі үлгі ретінде қолданамыз, оның тамаша жылдамдығы мен дәлдігі жүйенің өзегі ретінде. Әртүрлі сценарийлердегі қолданба талаптарын ескере отырып, біз пайдаланушыларға әртүрлі дәлдік пен жылдамдық талаптарына бейімделу үшін бірнеше таңдауды қамтамасыз ету үшін YOLOv7, v6 және v5 үлгілерін біріктіреміз.

Деректерді жақсарту : Модельдің әртүрлі қоршаған орта жағдайларында жалпылау қабілетін жақсарту үшін біз модельдің жауап беру қабілетін арттыру үшін кездейсоқ қиюды, айналдыруды, түсті өзгертуді және т.б. қоса, бірақ олармен шектелмей, деректерді жақсартудың әртүрлі әдістерін қолданамыз. жарықтандырудың өзгеруі, окклюзия және т.б. бейімделу.

Трансферттік оқыту : Алдын ала дайындалған YOLO үлгісін негіз ретінде пайдалана отырып, нақты қызанақ деректер жинағы трансферттік оқыту арқылы дәл реттеледі, осылайша оқыту процесін жылдамдатады және анықтау дәлдігін арттырады.

#### 2. Техникалық негіздерді және әзірлеу құралдарын таңдау

PyTorch құрылымы : PyTorch терең оқыту үшін негізгі жүйе ретінде таңдалған, оның икемділігі және қуатты GPU жеделдету мүмкіндіктері ол модельді оқыту мен қорытынды жасау процесін жылдамдатады.

Web UI үшін Streamlit : Streamlit қолданбасын интерактивті веб-бағдарлама интерфейсін құру үшін пайдаланыңыз, бұл пайдаланушыларға нақты уақытта

суреттерді, бейнелерді оңай жүктеп салуға және қызанақ піскенін анықтау нәтижелерін көруге мүмкіндік береді. Streamlit-тің қарапайымдылығы мен тиімділігі әзірлеу циклін айтарлықтай қысқартады және өнімнің прототиптерін тез жүзеге асырады.

PyCharm IDE : әзірлеу тиімділігі мен код сапасын жақсарту үшін PyCharm әзірлеу IDE ретінде пайдаланыңыз және оның бай әзірлеу құралдары мен қосылатын модуль экожүйесін пайдаланыңыз.

### 3. Жүйені жобалау және функцияны жүзеге асыру

Бірнеше кіріс көздерін қолдау : Жүйе әртүрлі пайдаланушы сценарийлерінің қажеттіліктерін қанағаттандыру үшін кескін файлдарын, бейне ағындарын және нақты уақыттағы камераны түсіруді қоса алғанда, бірнеше деректер енгізу көздеріне қолдау көрсетеді.

Модельді ауыстыру функциясы : Пайдаланушылар өздерінің қажеттіліктеріне сәйкес YOLOv8, v7, v6 немесе v5 пайдалануды таңдай алады. Жүйе пайдаланушы тәжірибесі мен жүйенің икемділігін арттыру үшін интерфейсте модельді ауыстыру функциясын қамтамасыз етеді.

Интерфейсті безендіру және өзара әрекеттесу дизайны : пайдаланушының өзара әрекеттесу тәжірибесін жақсарту үшін Streamlit әдепкі интерфейсін жақсарту үшін CSS пайдаланыңыз. Сонымен қатар, интерфейстің орналасуы интуитивті және оңай жұмыс істеуі қамтамасыз етілген, бұл пайдаланушыларға жылдам бастауды жеңілдетеді.

### 4. Мәліметтерді өңдеу және сақтау стратегиясы

Деректерді тиімді өңдеу : PyTorch деректерін алдын ала өңдеу және жүктеу механизмімен үйлескенде жүйенің жауап беру жылдамдығын қамтамасыз ету үшін деректерді тиімді өңдеу процесіне қол жеткізіледі.

Интеллектуалды деректерді сақтау : пайдаланушыларға тарихи деректерді сұрау және талдауды жеңілдету үшін анықтау нәтижелерін тиімді ұйымдастыру үшін негізделген деректерді сақтау шешімін жасаңыз.

### 5. Өнімділікті оңтайландыру және жүйені тестілеу

Өнімділікті реттеу : Жүйенің өнімділігін жүйелі түрде бағалаңыз, кедергілерді анықтаңыз және оңтайландырыңыз, үлгі күрделілігін азайтыңыз және үлгіні кесу және сандық анықтау сияқты техникалық құралдар арқылы жұмыс тиімділігін арттырыңыз.

Кешенді тестілеу : Жүйенің тұрақтылығы мен сенімділігін қамтамасыз ету үшін бірлікті тестілеуді, интеграциялық тестілеуді және өнімділікті сынауды қоса алғанда, кешенді жүйелік тестілеуді жүргізіңіз.

## 1.4 Деректер жиынын өңдеу

Бұл зерттеуде біз жалпы саны 6371 жоғары сапалы кескінді қамтитын қызанақтың піскендігін анықтауға арналған кәсіби деректер жинағын құруға назар аудардық. Бұл деректер жинағы піскен жасылдан піскен қызылға дейін

эртүрлі піскен қызанақтарды ғана емес, сонымен қатар зақымдалған қызанақ даналарын қамтиды. Мақсат - біздің модельдің нақты әлемде жалпылау қабілеті мен өнімділігіне мүмкіндік беру.

```
term_name = {"Green": "Жасыл", "Red": "ҚЫЗЫЛ", "damaged": "зақымданған"}
```

## 1.2-сурет – Деректер жинағы

Деректер жинағы мұқият үш бөлікке бөлінген: оқу жинағы, валидация жинағы және сынақ жинағы. Оқу жинағында 5520 кескін бар, ол эртүрлі жетілу деңгейлері бар қызанақтарды анықтауды үйрену үшін модельге негіз болады. Тексеру жинағында 566 кескін бар, олар шамадан тыс сәйкестендіруді болдырмау және модельдің дәл жалпылай алуын қамтамасыз ету үшін үлгі параметрлерін реттеу үшін пайдаланылады. Сынақ жинағы үлгінің соңғы өнімділігін бағалау үшін пайдаланылатын 285 кескінді қамтиды.

Деректерді енгізу үлгісінің жүйелілігі мен тиімділігін қамтамасыз ету үшін әрбір суретте стандартталған алдын ала өңдеу қадамдары орындалды. Біріншіден, біз пиксель деректерін автоматты түрде анықтаймыз және фотосурет жабдығы тудырған бағыттың ауытқуын жою үшін кескіннің EXIF ақпаратына сәйкес кескін бағытын реттейміз. Содан кейін барлық кескіндер 640x640 пиксель өлшеміне біркелкі реттелді және кескіндерді үлгі енгізу өлшемі талаптарына сай ету үшін созу әдісі қолданылды. Бұл өңдеу кескіннің бастапқы пропорцияларын сәл өзгертуі мүмкін болса да, біз әрбір суретте қызанақ әлі де анық көрінетініне көз жеткіздік.

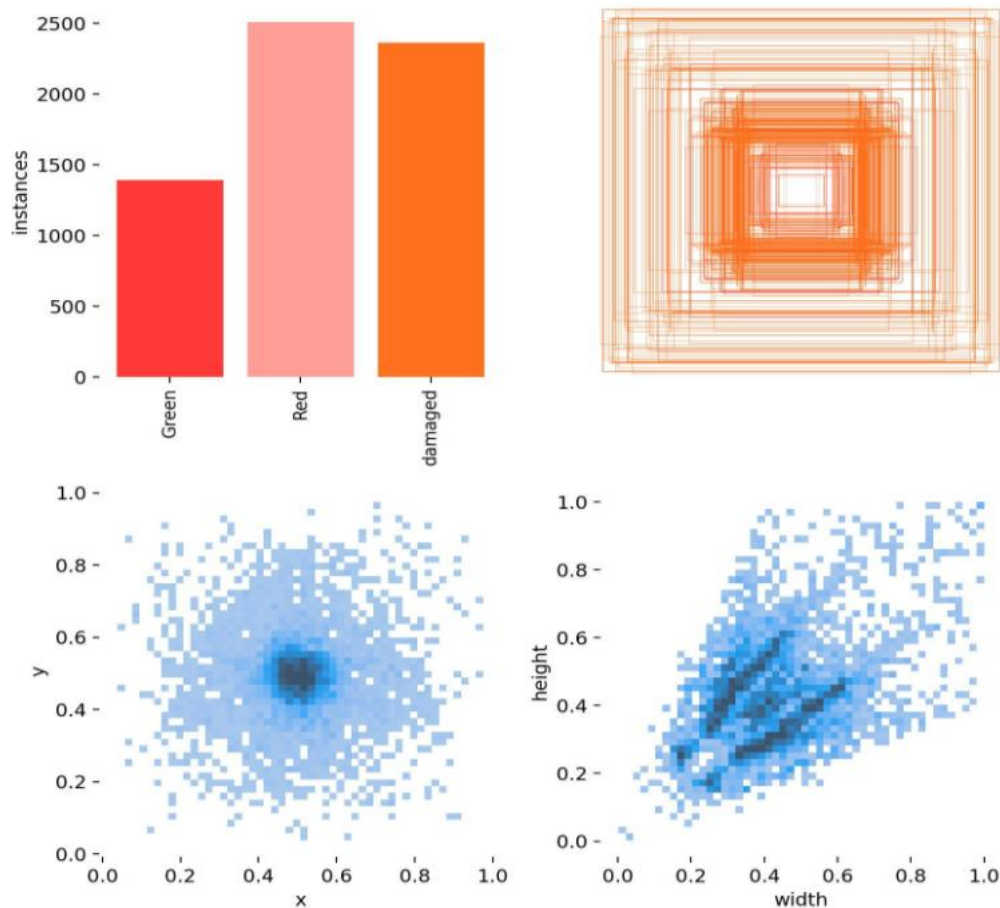
Таңбалау тұрғысынан біз екі тілді таңбалау жүйесін қабылдадық, ол әр санат үшін «Жасыл», «Қызыл» және «зақымдалған» сияқты ағылшын атауларын беріп қана қоймай, сонымен қатар сәйкес қытай атауларын берді: «Жасыл», «Қызыл» және «зақымданған». Бұл болашақта жоба командасының халықаралық ынтымақтастығын кеңейтуге көмектеседі, сонымен қатар қытайлық пайдаланушылардың түсінуі мен пайдалануын жеңілдетеді.





1.3-сурет – PyCharm платформасы арқылы YOLOv8-ді қолдану

Деректер жинағының таралуын талдау арқылы біз қызыл қызанақтардың ең көп суретке ие екенін, одан кейін зақымдалған қызанақтардың, ал жасыл қызанақтардың ең аз екенін анықтадық. Мұндай бөлу модельге нарықта жиі кездесетін піскен қызанақтарды жақсы анықтауға көмектеседі, сонымен қатар сатуға жарамсыз зақымдалған қызанақтарды анықтайды. Шектеу қораптарының таралуы қызанақтардың көпшілігі кескіннің орталық аймағында орналасқанын көрсетеді, бұл кескінді алу кезінде белгілі бір қалыпқа келтіру дәрежесі қолданылған болуы мүмкін екенін көрсетеді. Сонымен қатар, шектеу қорапшасының өлшемді таралуы салыстырмалы түрде шоғырланған, бұл деректер жинағындағы қызанақ мөлшері салыстырмалы түрде сәйкес келетінін көрсетеді, бұл модельге практикалық қолданбаларда қызанақ өлшемін үйренуге және болжауға көмектеседі.



1.4-сурет – Объектілерді анықтауға байланысты сыныптарды бөлу және визуализация даналарына шолу.

Қорытындылай келе, біздің деректер жинағы тек сандық және егжей-тегжейлі жіктеу бойынша жеткілікті емес, сонымен қатар мұқият алдын ала өңделген және аннотацияланған, YOLOv8/v7/v6/v5 негізіндегі қызанақтың пісуін анықтау моделін оқыту және бағалау үшін берік негіз береді. Мұндай деректер жинағы біздің зерттеуімізге тек академиялық үлес қосып қана қоймай, сонымен қатар практикалық қолдануда кең қолдану перспективаларына ие болады. Келесі тарауларда біз бұл деректерді үлгімізді үйрету үшін қалай пайдалану керектігін және модельдің әртүрлі өнімділік көрсеткіштері бойынша қалай жұмыс істейтінін қарастырамыз.

## 1.5 Принциптер мен кодекске кіріспе

### 1.5.1 YOLOv8 алгоритмінің принципі

Мақсатты анықтау саласындағы маңызды прогресс ретінде YOLOv8 алгоритмі принципі соңғы технологиялық инновацияларды және өнімділікті оңтайландыруды көрсетеді. Бұл модель дәстүрлі YOLO архитектурасын жақсартып қана қоймайды, сонымен қатар анықтау дәлдігі мен жылдамдығын жақсарту үшін бірқатар жаңа технологияларды енгізеді.

Ең алдымен, YOLOv8 нақтырақ желілік сәулет дизайнын енгізу арқылы модельдің шағын нысандарды тану қабілетін жақсартады, сонымен қатар фондық шуды басу мүмкіндігін жақсартады. Атап айтқанда, ол көп масштабты мүмкіндіктерді шығару технологиясын қабылдайды және әртүрлі өлшемдегі нысандардың ерекшеліктерін түсіре алады. Бұл мүмкіндік әсіресе қан жасушалары сияқты кішкентай нысандарды анықтау үшін өте маңызды, себебі олардың кескіндегі көрінісі өте кішкентай және оңай назардан тыс қалуы мүмкін. YOLOv8 әртүрлі деңгейлерде функцияларды біріктіру арқылы осы шағын нысандарды анықтау тиімділігін жақсарта алады.

Жоғалту функциясын жобалау кезінде YOLOv8 инновациялық «Тарату фокустық жоғалтуын» қабылдайды, бұл жіктеу қателерін жақсарту болып табылады. Дәстүрлі фокустық жоғалту негізінен жіктеу тапсырмаларындағы категория теңгерімсіздігі мәселесін шешуге арналған, ал «Тарату фокустық жоғалту» бір қадам алға жылжып, оны жіктеу ықтималдығының таралуын реттеу арқылы оңтайландырады. Жоғалту функциясының бұл жаңа түрі санаттар арасындағы айырмашылықтарды дәлірек көрсетіп қана қоймайды, сонымен қатар теңгерімсіз деректермен бетпе-бет келген кезде модельдің шамадан тыс сәйкестік құбылысын тиімді түрде азайта алады.

YOLOv8 сонымен қатар жаңа тапсырмаларды теңестіру тағайындау механизмі Task Aligned Assigner пайдаланады. Ол аннотация жолақтары мен болжанған жолақтар арасындағы теңестіруді бағалау арқылы нақты анкерлерге қандай аннотация жолақтарын тағайындау керектігін анықтайды. Task Aligned Assigner қолданбасын енгізу таңбалау мен болжау арасындағы қатені тиімді азайтады және үлгінің дәлдігін жақсартады.

Модельді оқыту процесінде YOLOv8 дизайнерлері оқу деректерінің сапасы мен тиімділігіне де ерекше көңіл бөлді. Бұл аспект үлгі өнімділігін жақсарту үшін оқу деректерін тиімдірек пайдалану жолында көрсетіледі. YOLOv8 әртүрлі ықтимал анықтау сценарийлерін модельдеу және күрделі орталарда модельдің жалпылау мүмкіндігін жақсарту үшін деректерді жақсарту технологияларының сериясын пайдаланады. Деректерді жақсарту технологиясын қолдану YOLOv8-ге әртүрлі деректерге неғұрлым тұрақты және сенімді оқу әсерлерін алуға мүмкіндік береді.

Қорытындылай келе, YOLOv8 желі архитектурасы, жоғалту функциясының дизайны, таңбалау қорабын бөлу механизмі және деректерді жақсарту

технологиясы сияқты көптеген аспектілерде жаңарған және оңтайландырылған. YOLOv8 жоғары дәлдіктегі анықтау нәтижелерін қамтамасыз етіп қана қоймайды, сонымен қатар нақты уақыттағы мақсатты анықтау жүйелерін енгізу және қолдану үшін күшті техникалық қолдау көрсете отырып, өңдеу жылдамдығы мен беріктігі бойынша жақсы жұмыс істейді.

## 1.5.2 Үлгі құрылысы

Біз терең техникалық талдауды қамтамасыз ету мақсатында YOLOv8/v7/v6/v5 негізіндегі қызанақтың піскендігін анықтау жүйесінің негізгі код бөлігін терең талқыладық. Терең оқыту және компьютерлік көру технологиясы арқылы бұл жүйе суреттердегі қызанақтарды дәл анықтап, жіктей алады және олардың жетілуін бағалай алады, бұл ауылшаруашылық өндірісі мен тамақ өңдеу өнеркәсібі үшін үлкен маңызға ие.

### Қажетті кітапханаларды импорттау

Біріншіден, код қажетті кітапханалар мен модульдерді импорттаудан басталады. cv2 бұл OpenCV кітапханасына сілтеме, ол негізінен әртүрлі кескіндер мен бейнелерді өңдеу тапсырмалары үшін пайдаланылатын ашық бастапқы компьютерлік көру және машиналық оқыту бағдарламалық қамтамасыз ету кітапханасы. torch Ол қуатты GPU жеделдетуін және икемді терең оқытуды зерттеу платформасын қамтамасыз ететін PyTorch, кеңінен қолданылатын ашық бастапқы компьютерлік оқыту кітапханасына қатысты.

```
# -*- coding: utf-8 -*-
import cv2
#Суреттерді өңдеуге арналған OpenCV кітапханасы
import torch # Pytorch Терең Оқыту Жүйесі
from QtFusion.models
import Detector, HeatmapGenerator
# QtFusion кітапханасы ұсынған мақсатты анықтау және жылу
картасын құру құралдары
from datasets.label_name import term_name
# Мақсатты анықтау санатының атауын Қытай Тіліне түрлендіру үшін
қолданылады
from ultralytics import YOLO
# Мақсатты анықтауға арналған Ultralytics YOLO моделі
from ultralytics.utils.torch_utils import select_device
# Қоршаған ортаға сәйкес есептеу техникасын таңдау үшін қолданылады
```

1.5-сурет – OpenCV кітапшасын ашу

Сонымен қатар, біз **QtFusion** және **ultralytics** кітапханаларының нақты сыныптары мен функцияларын енгіздік, олар мақсатты анықтауға және YOLO моделін енгізуге қолдау көрсетеді. `QtFusion.models` детектор және жылу картасын жасаушы кластары мақсатты анықтау тапсырмалары үшін инфрақұрылымды қамтамасыз етеді. `datasets.label_name` ішіндегі `term_name` сөздігі нәтижелердің оқылуын жақсарту үшін мақсатты санаттың ағылшынша атауын Қытай атауына түрлендіру үшін пайдаланылады. `ultralytics.YOLO` - бұл YOLO моделінің және `ultralytics.utils.torch_utils.select_device` функциясы есептеу тиімділігін арттыру үшін GPU немесе CPU сияқты оңтайлы есептеу құрылғысын автоматты түрде таңдау үшін пайдаланылады.

### Инициализация параметрлері және құрылғыны таңдау

Кодтың осы бөлігінде үлгі іске қосылатын құрылғыны (GPU немесе CPU), нысанды анықтауға арналған сенімділік шегін, максималды емес басу үшін IOU шегін (NMS) және санатты сүзу параметрлері. Бұл параметрлер анықтау процесінің дәлдігі мен тиімділігін қамтамасыз ете отырып, модельдің жұмысы үшін қажетті конфигурацияны қамтамасыз етеді.

```
device = "cuda:0" if torch.cuda.is_available() else "cpu"
ini_params = {
    'device': device,
    'conf': 0.25,
    'iou': 0.5,
    'classes': None,
    'verbose': False
}
```

1.6-сурет – Конфигурацияны ашу

### Санаттарды санау функциясы

`count_classes` функция — әр санаттың анықталған нысандарының санын есептейтін көмекші функция. Бұл функция анықтау ақпаратын және санат атауларының тізімін енгізу ретінде қабылдайды және санат атауларының тізіміне сәйкес нысан санауларының тізімін қайтарады. Бұл мүмкіндік сынақ нәтижелерін талдау және деректерді одан әрі өңдеу үшін пайдалы.

```

def count_classes(det_info, class_names):
    count_dict = {name: 0 for name in class_names}
    # Әр санаттың санын 0-ге дейін инициализациялаңыз
    for info in det_info: # Тест нәтижелерін өту
        class_name = info['class_name'] # Санат атауын алыңыз
        if class_name in count_dict: #Егер санат бар болса, санау 1-ге көбейтіледі
            count_dict[class_name] += 1
    count_list = [count_dict[name] for name in class_names]
    # Санау нәтижесін тізім формасына түрлендіру
    return count_list

```

### 1.7-сурет – Санаттарды санау нәтижесін еңгізу

#### YOLOv8v5Detector сыныбы

**YOLOv8v5Detector** Класс жүйенің өзегі болып табылады және ол **Detector** сыныптан мұраға алады және модельді жүктеу, кескінді алдын ала өңдеу, болжамды орындау және анықтау нәтижелерін кейінгі өңдеу сияқты негізгі функцияларды жүзеге асырады. Инициализация әдісінде модель мен параметрлер орнатылады. **load\_model** Әдіс алдын ала дайындалған YOLO үлгісін жүктеу және қажет болған жағдайда үлгі параметрлерін реттеу үшін қолданылады. **Preprocess** Әдіс үлгі енгізу талаптарын қанағаттандыру үшін кіріс кескінін өңдейді. **Predict** әдісі үлгі бойынша қорытынды жасайды және бастапқы анықтау нәтижелерін шығарады. Соңында, **postprocess** әдіс түпкілікті анықтау нәтижелерін шығару үшін NMS алгоритмін қолдануды, координат пішімдерін түрлендіруді және т.б. қоса алғанда, бастапқы анықтау нәтижелерін өңдейді.

Төменде келтірілген кодпен біз суреттердегі қызанақтарды дәл анықтай алатын және олардың піскендігін бағалай алатын қуатты анықтау жүйесін жасадық.

```

class YOLOv8v5Detector(Detector):
    def __init__(self, params=None):
        super().__init__(params)
        # Базалық класс конструкторына қоңырау шалыңыз
        self.model = None
        # Модельді бос күйге келтіріңіз
        self.img = None
        # Суретті бос күйге келтіріңіз
        self.names = list(term_name.values())
        # Барлық мақсатты санаттардың Қытайша атауларын алыңыз
        self.params = params if params else ini_params
        # Кіріс параметрлерін немесе әдепкі параметрлерді пайдаланыңыз

        def load_model(self, model_path): # Модельді жүктеу әдісі
            self.device = select_device(self.params['device'])
            # Есептеу құрылғысын таңдаңыз
            self.model = YOLO(model_path, ).to(self.device)
            # YOLO моделін жүктеп, көрсетілген құрылғыға өтіңіз
            names_dict = self.model.names
            # Модель санатының атауын алыңыз
            self.names = [Chinese_name[v] if v in Chinese_name else v for v in
names_dict.values()]

            # Модельді жылыту
            self.model(torch.zeros(1, 3, 640, 640).to(self.device).
type_as(next(self.model.model.parameters())))

        def preprocess(self, img): # Кескінді алдын ала өңдеу
            self.img = img # Түпнұсқа суретті сақтаңыз
            return img # Бастапқы кескінге оралу

```

1.8-сурет – Өңдеу функциясы

### 1.5.3 Зерттеудің коды

Кесте 1 - Гиперпараметрлердің сипаты

гиперпараметрлер	сипаттау	Мысал мәні
epochs	Модельді оқытудың жалпы айналым саны, яғни деректер жиыны қанша рет толық оқытылады. Дәуірлердің көп саны үлгі өнімділігін жақсартуы мүмкін, бірақ сонымен бірге шамадан тыс орнатуға әкелуі мүмкін.	120
batch	Әрбір жаттығу итерациясында пайдаланылатын үлгілер саны. Үлкенірек пакет өлшемдері жаттығудың тұрақтылығы мен тиімділігін жақсартады, бірақ жад талаптарын арттырады.	16
imgsz	Кескіннің өлшемін енгізіңіз. YOLO моделі барлық кескіндердің өлшемдері бірдей болуын талап етеді. Әдетте үлкенірек кескін өлшемін таңдау үлгінің дәлдігін жақсартады, бірақ есептеу жүктемесін арттырады.	640
iou	Максималды емес басу (NMS) үшін біріктіру (IoU) шегінің қиылысы. Қайталанатын анықтау жақтауларын жою және ең жақсы анықтау нәтижелерін сақтау үшін қолданылады.	0,7
lr0	Бастапқы оқу жылдамдығы. Оқу жылдамдығы салмақты реттеудің шамасын басқарады. Оқыту жылдамдығының жоғары болуы жаттығуды тездетуі мүмкін, бірақ сонымен бірге тұрақсыз жаттығуларға әкелуі мүмкін.	0,01
momentum	Оңтайландырушы импульс. Импульс сәйкес бағыттар бойынша SGD конвергенциясын жеделдетуге көмектеседі және тербелістерді азайтады.	0,937
weight_decay	Салмақ ыдырауы (L2 реттелуі). Үлкен салмақтарды жазалау үшін жоғалту функциясына реттеу терминін қосу арқылы үлгінің шамадан тыс қиын болдырмауға көмектеседі.	0,0005
augment	Деректерді кеңейтуді пайдалану керек пе. Деректерді көбейту үлгінің жалпылау қабілетін жақсартуға көмектесетін оқу кескіндеріне кездейсоқ түрлендірулер сериясын қолдану арқылы деректердің әртүрлілігін арттырады.	жалған



Блогтың осы бөлігінде біз мақсатты анықтау моделін оқытудың негізгі кодтық процесін біртіндеп талдап, оның құрылымы мен функцияларын ашып, оны оқытудың нақты тапсырмаларына қалай қолдану керектігін талқылаймыз. Келесі кестеде YOLOv8 үлгісін оқытуда қолданылатын кейбір маңызды гиперпараметрлер және олардың параметрлері келтірілген:

Біз YOLOv8 үлгісін қызанақтың піскенін анықтау моделін үйрету үшін қалай пайдалану керектігін қарастырамыз. Арнайы код үзінділерін біріктіру арқылы біз үлгіні оқытудың әрбір негізгі қадамын егжей-тегжейлі талдаймыз және нақты техникалық нұсқаулықты береміз.

### **Қажетті кітапханалар мен модульдерді импорттаңыз**

Алдымен біз қажетті кітапханалардан тиісті модульдерді импорттадық. Модульдер os файлдық және каталогтық операциялар үшін пайдаланылады. YAML файлдарын өңдеуге арналған кітапхана, конфигурация файлдары үшін әдетте пайдаланылатын деректерді сериялау пішімі. Соңында біз YOLOv8 үлгісін енгізудің кілті болып табылатын torchyamlultralyticsYOLO пакеттен сыныпты импорттадық.

```
import os
import torch
import yaml
from ultralytics import YOLO
from QtFusion.path import abs_path
```

### 1.9-сурет – YOLO моделін импорттау

#### **Есептеу құрылғысын орнату**

Модельді оқытуға кіріспес бұрын, оқыту процесі қандай құрылғыда орындалатынын анықтауымыз керек. Мұнда қол жетімді CUDA құрылғысының (яғни GPU) бар-жоғын анықтау үшін қарапайым шартты өрнекті қолданамыз. Егер бар болса, біз бірінші CUDA құрылғысын қолданамыз, әйтпесе, CPU әдепкі бойынша пайдаланылады;

```
device = "0" if torch.cuda.is_available() else "cpu"
```

### 1.10-сурет – Есептеу құрылғысын орнату

#### **Жаттығу деректерін дайындаңыз**

Әрі қарай, біз оқытуға қажетті кейбір негізгі параметрлерді, соның ішінде жұмысшы процестерінің санын және партия өлшемін анықтаймыз. Біз сонымен қатар оқу деректер жинағының атауы мен жолын көрсеттік. Функция арқылы abs\_path біз деректер жиынының конфигурация файлына абсолютті жолды аламыз және оны платформааралық үйлесімділік үшін UNIX стиліндегі жолға түрлендіреміз.

```
workers = 1
batch = 8
data_name = "TomatoRipenessDet"
data_path = abs_path(f'datasets/{data_name}/{data_name}.yaml',
path_type='current')
unix_style_path = data_path.replace(os.sep, '/')
```

### 1.11-сурет – Жаттығу деректерін дайындау

#### YAML конфигурация файлын өзгертіңіз

Жаттығу деректеріне жол дұрыс екеніне көз жеткізу үшін алдымен деректер жиынының конфигурация файлының мазмұнын оқып, қажет болған жағдайда оны өзгертеміз. Мұнда біз конфигурациядағы жазбаны тексеріп **path**, оны деректер жинағы орналасқан каталогқа апаратын жолға жаңарттық. Бұл қадам оқу процесінің барлық оқу деректеріне дұрыс қол жеткізуін қамтамасыз етеді.

```
directory_path = os.path.dirname(unix_style_path)with open(data_path, 'r') as file:
    data = yaml.load(file, Loader=yaml.FullLoader)if 'path' in data:
    data['path'] = directory_path
with open(data_path, 'w') as file:
    yaml.safe_dump(data, file, sort_keys=False)
```

### 1.12-сурет – YAML конфигурация файлын өзгерту

#### Үлгіні жүктеп, жаттығуды бастаңыз

Соңында біз алдын ала дайындалған YOLOv8 үлгісін жүктеп, оқу процесін бастадық. көрсету арқылы сынып данасына біздің мақсатымыз нысанды анықтау екенін **task='detect'** айтамыз. YOLO Әдісте **model.train** біз деректер конфигурациялау файлына жолды, есептеу құрылғысын, жұмысшы процестерінің санын, кескін өлшемін, оқыту дәуірлерінің санын, партия өлшемін және атын қамтитын бірқатар параметрлерді береміз. жаттығу тапсырмасы. Бұл параметрлер бірге оқу процесінің тәртібін анықтайды.

Төменде келтірілген код арқылы біз YOLOv8 үлгісіне негізделген қызанақтың пісуін анықтау үлгісін оқытуды орнату және орындау процесін аяқтадық. Бұл процесс деректерді дайындау мен конфигурациялауды реттеуді ғана емес, сонымен қатар модельді жүктеуді және оқытуды орындауды қамтиды.

```

model = YOLO(abs_path('./weights/yolov8n.pt'), task='detect')
results2 = model.train(
    data=data_path,
    device=device,
    workers=workers,
    imgsz=640,
    epochs=120,
    batch=batch,
    name='train_v8_' + data_name)

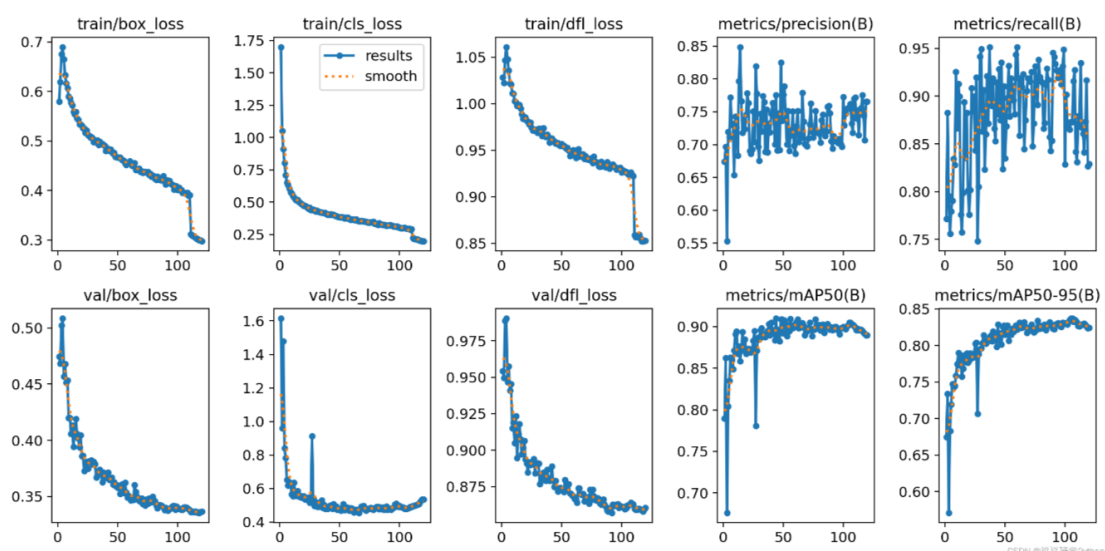
```

1.13-сурет – Үлгіні жүктеу

## 2 Эксперимент нәтижелері және талдау

### 2.1 Зерттеу графиктері

Жаттығу процесі кезінде YOLOv8 моделінің жоғалту функциясының бейнесін терең талдаймыз. Бұл диаграммалар модельді оқыту өнімділігінің негізгі көрсеткіштері болып табылады және деректерден үйрену барысында модельдің қалай орындалатынын және ілгерілеуін түсінуге көмектеседі.



2.1-сурет – Қызанақтың пісу деңгейлерін санау және жіктеу Үшін YOLOv8 моделін оқытудан кейінгі орташа Дәлдікті (mAP) және жоғалту тенденцияларын визуализациялау

Тәжірибе нәтижелерін талдай отырып, біз YOLOv5nu 0,911 mAP ұпайымен көш бастайтынын байқадық, бұл оның әр түрлі пісетін қызанақтарды орналастыру және жіктеудегі жоғары дәлдігін көрсетеді. Кешенді индекс ретінде mAP модельдің өнімділігін әртүрлі сенімділік шегінде бағалайды. YOLOv5nu жоғары mAP оның желі құрылымын оңтайландыруға және мүмкіндіктерді шығару мен шектейтін қорапты болжаудың ықтимал тиімділігін арттыруға байланысты.

YOLOv7-tiny F1-Score бойынша 0,83 тамаша өнімділікке ие. Дәлдік пен еске түсірудің гармоникалық орташа мәні ретінде, F1-Score нақты қолданба сценарийлеріндегі үлгі өнімділігіне сұранысқа жақынырақ. YOLOv7-tiny жоғары еске түсіру жылдамдығын қамтамасыз ете отырып, салыстырмалы түрде жоғары дәлдікті сақтайды, бұл модель өткізіп алған анықтауларды азайта отырып, жалған анықтауларды азайта алатындығын көрсетеді, бұл осындай тапсырмалар үшін жетілуді жіктеу сияқты дәл орындауды қажет ететін қолданбалар үшін пайдалы өте маңызды.

YOLOv6n және YOLOv8n көрсеткіштері mAP және F1-Score бойынша сәл төмен болғанымен, олардың өнімділігі әлі де күшті. Екі нұсқада әртүрлі аспектілерде келіссөздер жасалған болуы мүмкін, мысалы, YOLOv8n алгоритмнің беріктігі мен жылдамдығын оңтайландырған болуы мүмкін, осылайша модель практикалық қолданбаларда жылдамдық пен дәлдік арасындағы тепе-теңдікті сақтай алады.

Біздің эксперименттік нәтижелеріміз объектіні анықтау моделін таңдау кезінде нақты тапсырмалардың қажеттіліктерін толығымен ескеру керектігін көрсетеді. Мысалы, мақсат жылдамырақ өндіріс желісінде жылдам сұрыптауды орындау болса, сізге жылдамдықта артықшылығы бар үлгіні таңдау қажет болуы мүмкін, мысалы, сапаны тексеру немесе зиянкестер мен ауруларды анықтау; содан кейін mAP жоғары үлгілерге басымдық берілуі керек.

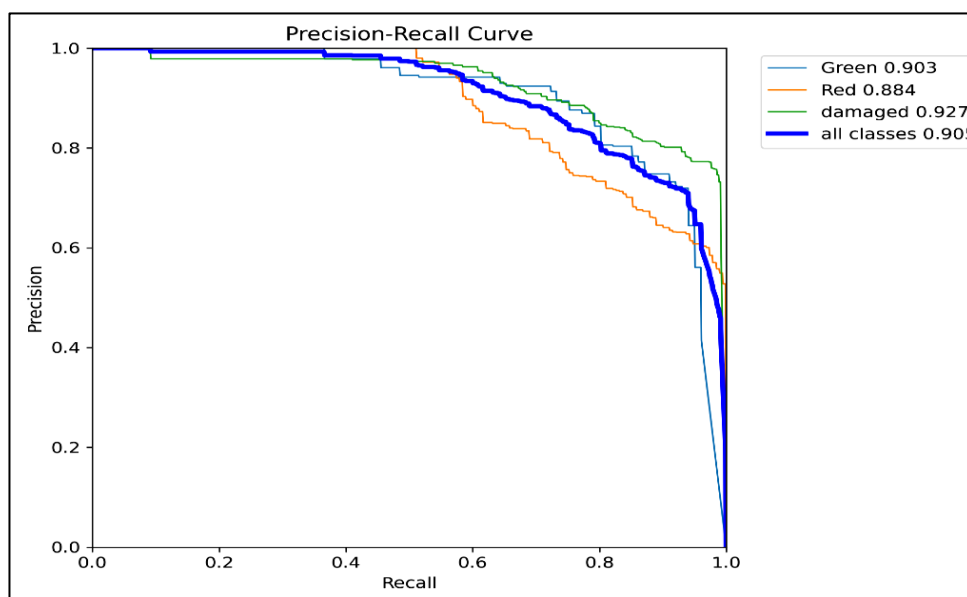
Қорытындылай келе, әртүрлі YOLO нұсқалары қызанақ піскенін анықтау тапсырмасында әртүрлі артықшылықтарды көрсетеді. Осы үлгілерді терең талдау арқылы блогер ауыл шаруашылығын автоматтандыру саласындағы технологияны таңдауға анықтама береді және болашақ зерттеулердің бағытын көрсетеді, яғни талаптарға бейімделу үшін модель құрылымы мен оқыту үдерісін одан әрі оңтайландыру жолын көрсетеді. нақты қолдану сценарийлеріндегі жылдамдық пен дәлдік.

## **PR қисығы**

Томаттың піскендігін анықтау үшін YOLOv8 үлгісін пайдаланғаннан кейін алынған Precision-Recall (PR) қисығын егжей-тегжейлі түсіндірейік. PR қисығы классификатордың өнімділігін өлшеудің маңызды құралы болып табылады, әсіресе деректер жинағы класстық теңгерімсіз болған кезде. Қисық сызығының әрбір нүктесі белгілі бір қайта шақыру шегінде жіктеуіштің дәлдігін көрсетеді. Ең дұрысы, біз классификатордың еске түсірудің барлық деңгейлерінде жоғары

дәлдікті сақтауын қалаймыз, бұл PR қисығын координаттардың жоғарғы оң жақ бұрышын құшақтайтындай етеді.

Біздің жағдайда PR қисығы қызанақтың үш түрін анықтау өнімділігін көрсетеді: «Жасыл», «Қызыл» және «зақымдалған» қызанақ. «Жасыл» категориясының mAP (орташа дәлдік) мәні 0,903, «Қызыл» категориясы 0,884, «зақымдалған» категориясы 0,927-ге жеткенін көруге болады. Бұл модельдің «зақымданған» қызанақтарды жақсы анықтайтынын білдіреді, себебі бүлінген қызанақтардың айқынырақ сипаттамалары бар, бұл модельге оларды ажыратуды жеңілдетеді. «Жасыл» және «Қызыл» қызанақтар үшін mAP мәні сәл төмен болса да, ол әлі де 0,9 шамасында, бұл модельдің жетілуді бағалауда да жоғары дәлдікке ие екенін көрсетеді.



2.2-сурет – Дәл көбею графигі.

Тұтастай алғанда, барлық санаттардың mAP мәні 0,905-ке жетті, бұл біздің модельдің әртүрлі жетілу дәрежесі бар қызанақтарды анықтаудың өте жақсы өнімділігін көрсетеді. Сонымен қатар, бүкіл PR қисығын бақылай отырып, біз еске түсіру жылдамдығы төмен болған кезде дәлдік өте жоғары деңгейде сақталатынын және еске түсіру жылдамдығы жоғарылаған сайын дәлдіктің біртіндеп төмендейтінін таба аламыз. Бұл тенденция жіктеу тапсырмаларында жиі кездеседі, өйткені біз үлгідегі талаптарды босаңсытып, оған көбірек оң үлгілерді анықтауға мүмкіндік бергенде, жалған позитивтер саны артады.

Ескерту 1-ге жақындаған кезде барлық қисық сызықтар айқын төмендеуді көрсететінін атап өткен жөн. Бұл жалған позитивтердің (жалған позитивтердің) артуына байланысты барлық дерлік оң үлгілер анықталған жағдайларда дәлдіктің төмендеуін көрсетеді. Дегенмен, жалпы өнімділіктің төмендеуі жоғарырақ дәлдік деңгейінде басталады, бұл модель салыстырмалы түрде берік екенін көрсетеді.

Осы PR қисығы арқылы біз YOLOv8 моделі әртүрлі жетілу деңгейлері бар қызанақты анықтау тапсырмаларын орындауда жақсы жұмыс істейді және көптеген нысандарды жоғары дәлдікпен анықтай алады деген қорытындыға келуге болады. Әсіресе «зақымдалған» санат үшін модельдің өнімділігі ерекше. Бұл бізге ауыл шаруашылығын автоматтандыру саласында қолданудың қуатты құралын ұсынады, ол фермерлерге әртүрлі штаттардағы қызанақтарды тиімдірек жіктеуге және өңдеуге, бүкіл саланың тиімділігі мен өнім сапасын арттыруға көмектеседі.

## 2.2 YOLOv8/v7/v6/v5 тәжірибесін салыстыру

### Эксперименттік дизайн :

Бұл эксперимент мақсатты анықтау тапсырмаларындағы YOLOv5, YOLOv6, YOLOv7 және YOLOv8 үлгілерінің өнімділігін бағалауға және салыстыруға бағытталған. Осы мақсатқа жету үшін төрт модель бірдей деректер жиынын пайдаланып, өнімділікті тікелей салыстыруға мүмкіндік беретін жеке оқытылды және сынақтан өтті. Бұл деректер жинағы нысандардың кескіндерін қамтиды. Бұл мақалада әр модельдің артықшылықтары мен кемшіліктерін ашуға және оларды өндірістік ортада практикалық қолдану үшін сценарийлерді таңдауға бағытталған төрт модель салыстырмалы түрде талданады.

Кесте 2.1 – YOLO үлгілерін өзара параметрлерін салыстыру

Үлгі	Кескін өлшемі (пиксель)	мАПвал 50-95	CPU ONNX жылдамдығы (мс)	A100 TensorRT жылдамдығы (мс)	Параметрлер саны (млн)	FLOPs (млрд)
YOLOv5nu	640	34.3	73.6	1.06	2.6	7.7
YOLOv8n	640	37.3	80.4	0,99	3.2	8.7
YOLOv6N	640	37.5	-	-	4.7	11.4
YOLOv7 кішкентай	640	37.4	-	-	6.01	13.1

### Метрикалық көрсеткіштер :

F1-Score: F1-Score модель өнімділігін өлшеу үшін маңызды көрсеткіш болып табылады, әсіресе санаттардың біркелкі бөлінбеген деректер жиынымен жұмыс істегенде. Дәлдік пен еске түсіруді біріктіре отырып, ол үлгі өнімділігін жан-жақты бағалай алатын жалғыз метриkanı қамтамасыз етеді. Дәлдік модель анықтай алатын барлық белгіленген позитивтердің арасындағы шынайы позитивтердің үлесін өлшейді, ал еске түсіру модель барлық нақты позитивтердің арасында анықтай алатын шынайы позитивтердің үлесіне назар

аударды. F1-Score модельдің өнімділігін бағалау дәлдік жылдамдығы мен еске түсіру жылдамдығы екеуі де жоғары болған кезде ғана жоғары болатынына көз жеткізу үшін екеуінің орташа гармоникалық мәнін пайдаланады, осылайша модельдің оң мысалдарды болжауы дәл әрі толық болуын қамтамасыз етеді.

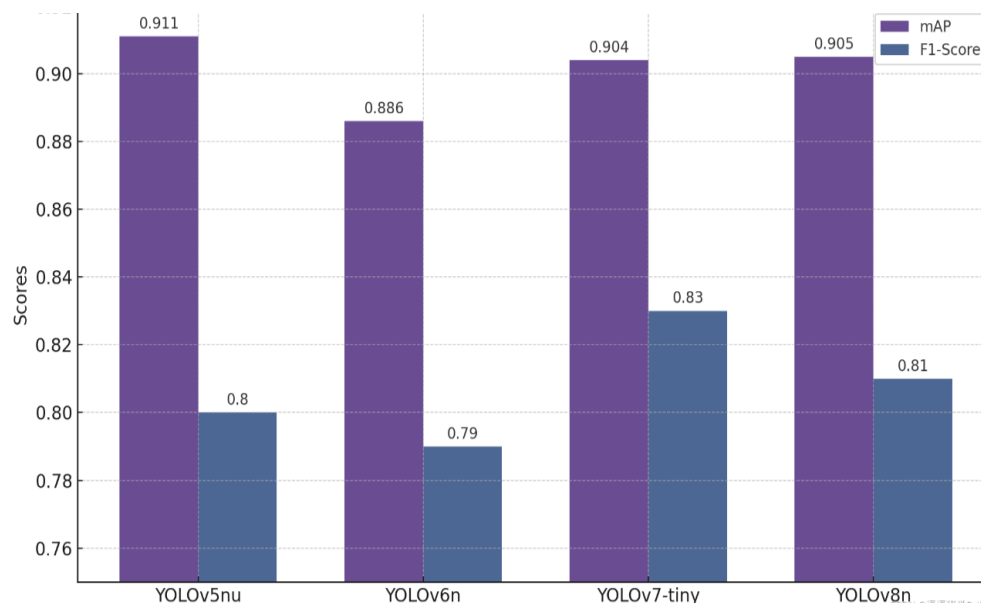
mAP (Орташа орташа дәлдік): Мақсатты анықтау тапсырмаларында орташа орташа дәлдік (mAP) үлгі өнімділігін бағалаудың маңызды критерийі болып табылады. Ол бір санат үшін үлгінің тану дәлдігін көрсетіп қана қоймайды, сонымен қатар барлық санаттардың орташа өнімділігін ескереді, осылайша жаһандық өнімділік өлшемін береді. mAP есептеу кезінде әрбір санат үшін модельдің болжамдары жеке қарастырылады, содан кейін әрбір санат үшін орташа дәлдік (AP) есептеледі және соңында осы AP мәндерінің орташа мәні mAP құрайды.

Кесте 2.2 – YOLO платформасын бағалау

Аты	YOLOv5nu	YOLOv6n	YOLOv7- кішкентай	YOLOv8n
<b>mAP</b>	0,911	0,886	0,904	0,905
<b>F1-үпай</b>	0,80	0,79	0,83	0,81

#### Эксперимент нәтижелерін талдау :

Бұл бөлімде біз YOLOv5nu, YOLOv6n, YOLOv7-tiny және YOLOv8n төрт моделінің бір деректер жиынындағы өнімділігін терең талдау және салыстыру жүргіземіз. Біз эксперименттің негізі мен мақсатымен таныстырамыз, содан кейін модельдер арасындағы өнімділік көрсеткіштерін салыстырамыз және соңында эксперимент нәтижелерінің егжей-тегжейлі талдауын береміз.



2.3-сурет – Деректер жиынындағы YOLO нұсқаларын салыстыру

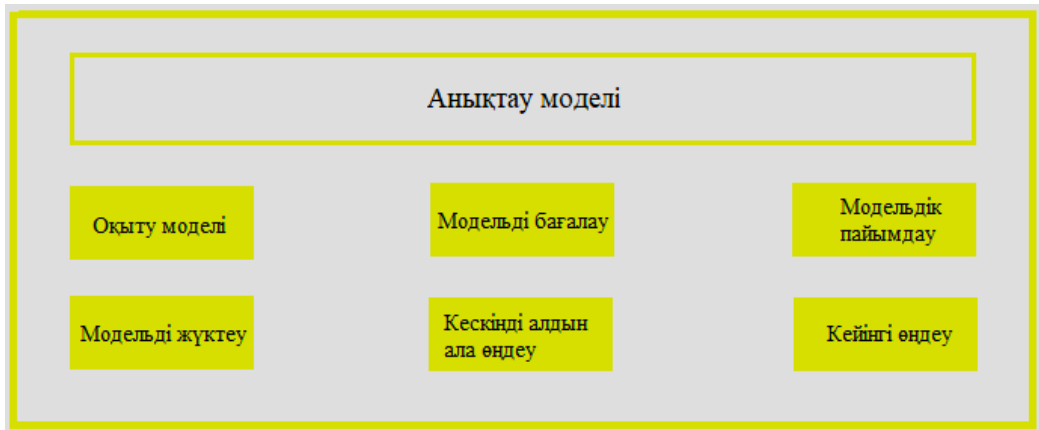
### 2.3 Жүйені жобалау

Бұл блогта біз YOLO алгоритмдер сериясына негізделген мақсатты анықтау жүйесінің **жүйелік архитектурасын** терең талдаймыз. Біздің концепция жұмыс істеуге оңай, тиімді, дәл және жақсы пайдаланушы тәжірибесі бар жүйені құру болып табылады. Жүйе күндіз-түні әртүрлі мақсатты ақпаратты тез анықтап, жаза алады. Төмендегілер жүйе архитектурасының негізгі құрамдас бөліктері болып табылады:



2.4-сурет – Жүйелік архитектурасы





2.5-сурет – Анықтау моделі

**Модельді жүктеу және алдын ала өңдеу:** Жүйенің архитектуралық өзегі- **YOLOv8v5Detector** класы. Бұл сыныпта YOLO моделінің алдын-ала дайындалған параметрлері қолданылады (usually.pt файл), мақсатты тану тапсырмаларын инициализациялау және орындау. **YOLOv8v5Detector** кескінді өңдеудің және қорытындыларды болжаудың бүкіл процесін ішкі түрде инкапсуляциялайды. \* \* Load\_model \* \* әдісі модельдің кейінгі анықтау тапсырмаларына дұрыс қолданылуын қамтамасыз ету үшін модель салмақтарын жүктеуге жауап береді.

**Конфигурацияны басқару:** пайдаланушы интерфейсінің өзара әрекеттесуін бүкіл жүйенің пайдаланушылық өзара әрекеттесу логикасын біріктіретін **Detection\_UI** класы басқарады. Бүйірлік тақта конфигурациясы арқылы пайдаланушылар анықтау дәлдігі мен сезімталдығын реттеу үшін үлгі параметрлерін (соның ішінде **\*\*model\_type, conf\_threshold және iou\_threshold**) дербес орната алады. **Сондай-ақ пайдаланушылар өздерінің үлгі файлдарын жүктей алады және жүйе load\_model\_file\*\*** әдісі арқылы анықтау үшін осы реттелетін үлгілерді жүктеп алып, пайдаланады.

**Кескінді және бейнені өңдеу :** Әртүрлі кіріс көздері үшін - камералар, сурет файлдары немесе бейне файлдары үшін **Detection\_UI** сыныптағы әдістер **process\_camera\_of\_file** осы кірістерді өңдеуге жауапты. Бұған камералардан тірі кескіндерді түсіру, жүктеп салынған файлдарды оқу және декодтау және тану үшін үлгілерді шақыру кіреді.

**Нәтижені көрсету және тіркеу :** Анықтау нәтижелерін жазу және көрсету **ResultLogger** және **LogTable** сыныптары арқылы жүзеге асырылады. **ResultLogger** сыныбы анықтау нәтижелерін нақты уақытта жаңарту және көрсету үшін пайдаланылады, ал **LogTable** сыныбы пайдаланушыларға тарихи анықтау деректерін сақтауға және қарауға мүмкіндік беретін нәтижелер үшін тұрақты сақтау мүмкіндіктерін қамтамасыз етеді.

**UI дизайны :** бүкіл жүйе дизайнында біз анықтау нәтижелерін тануды жақсарту үшін түстерді кездейсоқ бөлу стратегиясын да енгіздік. Жүйе әрбір

анықталған санатқа динамикалық түрде түс тағайындайды, бұл процесс **Detection\_UI** класындағы **\*\*colors\*\*** атрибуты арқылы басқарылады .

**Нақты уақыттағы жаңартулар және кері байланыс** : Жүйе орындалу барысы мен динамикалық жаңарту механизмімен жасалған `st.progress` және `st.image` сияқты Streamlit құрамдастары арқылы модельді өңдеу барысы мен нәтижелері туралы нақты уақыттағы кері байланыс пайдаланушының интерактивті тәжірибесін жақсартады.

## 2.4 Жүйелік процесс

YOLOv8/v7/v6/v5 негізіндегі мақсатты анықтау жүйемізде барлық анықтау процесі күрделі дизайн идеялары мен пайдаланушы тәжірибесін терең түсінуді көрсетеді. Төменде біз осы жүйе процесінің әрбір қадамын бағдарламаның ағын диаграммасы түрінде егжей-тегжейлі таныстырамыз.

### 1. Жүйені инициализациялау :

**Жүктеу моделі**: жүйе іске қосылғанда, `load_model` әдісі YOLOv8v5Detector класының инстанциясы кезінде оқытылған YOLO моделінің салмақтарын жүктеу үшін шақырылады.

**Кездейсоқ түстерді генерациялау** : әртүрлі мақсатты санаттарды ажырату үшін жүйе әр санатқа кездейсоқ түстерді тағайындайды және оларды `colors` массивінде сақтайды.

### 2. Интерфейс параметрлері :

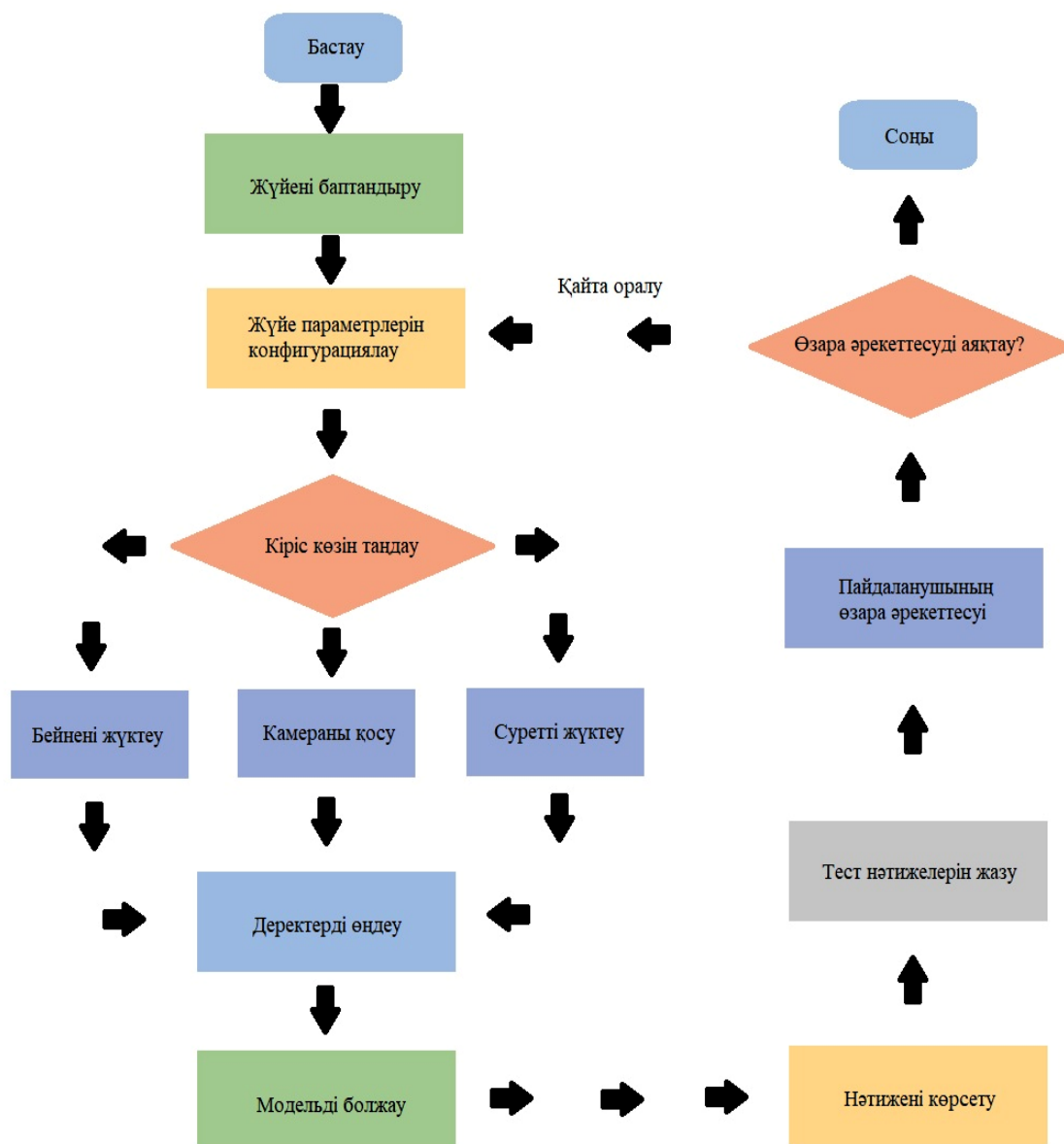
**Бет конфигурациясы** : `setup_page` әдіс арқылы бет макетін және тақырыбын конфигурациялаңыз.

**Бүйірлік тақта конфигурациясы** : Бүйірлік тақтада `setup_sidebar` үлгі параметрлері, сенімділік және IOU шектері үшін реттеу сырғытпаларын қамтамасыз ету үшін әдісті пайдаланыңыз.

### 3. Пайдаланушы әрекеттесуі :

**Файлды жүктеп салу** : Пайдаланушылар реттелетін сурет немесе бейне файлдарын жүктей алады немесе тікелей камера түсірген кадрларды таңдай алады.

**Үлгіні таңдау** : Пайдаланушылар әдепкі үлгіні пайдалануды немесе пайдаланушы үлгі файлы жүктеп салуды таңдай алады.



2.6-сурет – Орындау алгоритмі

#### 4. Анықтаудың орындалуы:

**Кіріс көзін өңдеу** : пайдаланушы таңдауына негізделген process\_camera\_or\_file әдіс камерадан тікелей түсірілген бейнені немесе жүктеп салынған файлдарды өңдеу керектігін анықтайды.

**Кескінді алдын ала өңдеу** : үлгінің енгізу талаптарына сәйкес келетін кескіндердің өлшемін өзгертіңіз және басқа қажетті кескінді өңдеу қадамдарын орындаңыз.

**Модельді болжау** : Анықтау нәтижелерін алу үшін алдын ала өңделген кескінді YOLO үлгісіне енгізіңіз.

## 5. Нәтижелерді көрсету және жазу :

**Анықтау нәтижесінің дисплейі :** frame\_process әдісі әрбір кадрдың анықтау нәтижелерін көрсетеді, соның ішінде шектеу жолақтарын салу және белгілерді көрсету.

**Нәтижені жазу :** ResultLogger сынып анықтау нәтижелерін жазады және LogTable сыныпты пайдаланып нәтижелерді CSV файлына сақтайды.

## 6. Тұтынушылардың кері байланысы :

**Нәтижелерді сүзу және көрсету :** пайдаланушылар белгілі бір мақсаттарды сүзу үшін бүйірлік тақтадағы ашылмалы мәзірді пайдалана алады және жүйе toggle\_comboBox әдісі арқылы таңдалған мақсаттың толық ақпаратын көрсетеді.

**Динамикалық нәтижені жаңарту :** Жүйе анықтау нәтижелерін нақты уақытта жаңартады және оларды интерфейсте ұсынады.

## 7. Жүйе аяқталады :

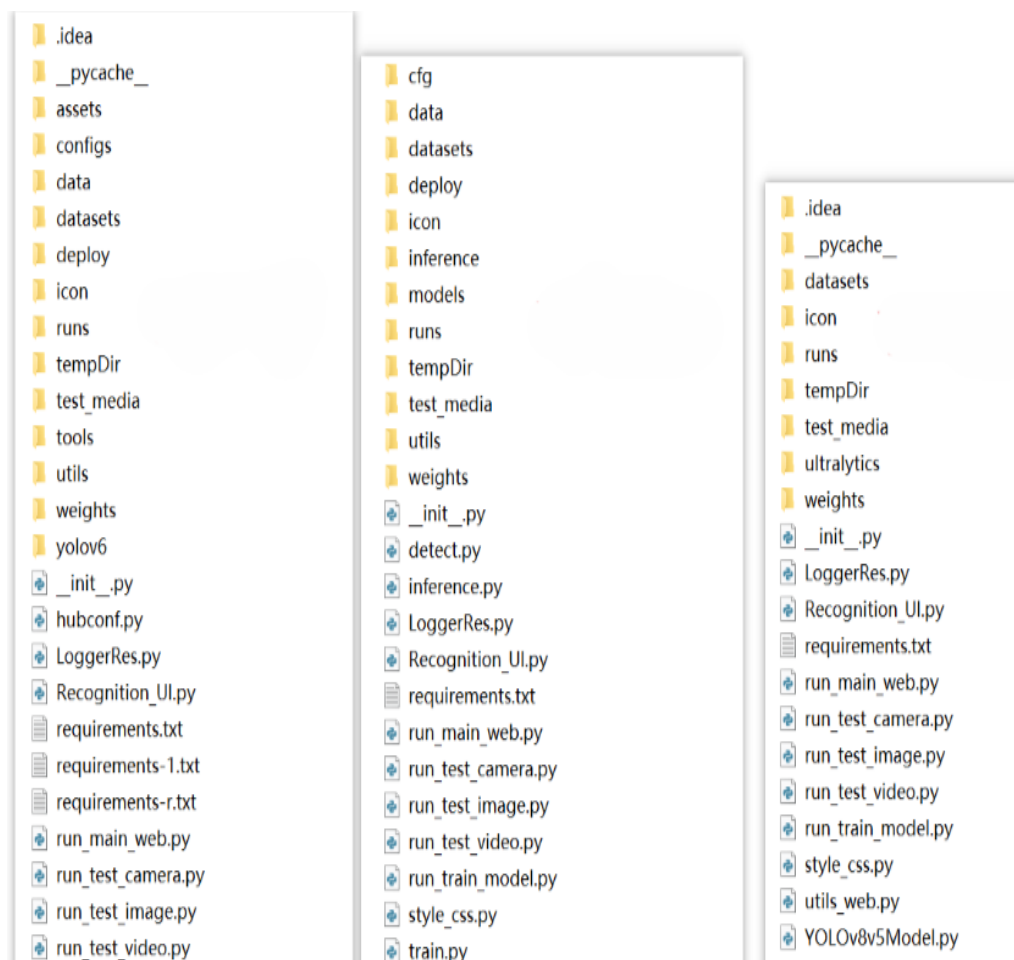
**Анықтауды тоқтату :** Пайдаланушылар анықтау процесін кез келген уақытта «Тоқтату» түймесін басу арқылы аяқтай алады.

**Журналды сақтау :** LogTable сынып жүйе аяқталған кезде барлық анықтау журналдарын сақтайды және экспорттау функциясын қамтамасыз етеді.

Бұл анықтау процесінің дизайны пайдаланушы жұмысының ыңғайлылығын, жүйенің нақты уақыт режимінде жауап беруін және нәтижелерді дәл жазуды толығымен қарастырады. Техникалық тереңдік немесе пайдаланудың қарапайымдылығы тұрғысынан біз пайдаланушыларға қанағаттанарлық тәжірибе беруге тырысамыз. Мұндай процесті жобалау арқылы ол әртүрлі мақсаттарды жылдам тауып, анықтап қана қоймай, сонымен қатар пайдаланушыларға деректерді кейінгі талдау және басқару шешімдерінде көмектесу үшін пайдаланушыларға егжей-тегжейлі анықтау жазбаларын қалдыра алады.

## Кодты жүктеу сілтемесі

Блогта айтылған **толық ресурс пакетін**, соның ішінде сынақ суреттерін, бейнелерді, Python файлдарын (\*.py), веб-конфигурация файлдарын, оқу деректер жиынын, кодты және интерфейс дизайнын және т.б. алғыңыз келсе, блогердің веб-сайтына кіре аласыз. Нандағы көп платформалы жүктеп салу мазмұны. Байланысты блог және бейне материал бір рет басу әрекеті үшін барлық қажетті файлдарға жүктеу сілтемелерін береді. Толық ресурстың алдын ала қарауы төменде көрсетілген:



2.7-сурет – YOLO-дағы кодты Python-ға жүктеу

Ресурстар пакеті оқыту және тестілеу деректер жинақтары, оқыту және тестілеу коды, UI интерфейс коды және т.б. сияқты толық ресурстарды қамтиды.

### **Толық орнату және іске қосу оқулығы:**

Бұл жобаның жұмысы үшін Anaconda және Pycharm пайдалану қажет. Ресурс кодын жүктеп алғаннан кейін оны сәтті іске қосу үшін сізде әлі де болса, жеке хабарлама жіберуге болады оны шешу үшін блогер:

#### **1. Pycharm және Anaconda үшін орнату оқулығы:**

<https://deepcode.blog.csdn.net/article/details/136639378> ;

Бағдарламалық құрал орнатылғаннан кейін осы жоба үшін жаңа Python ортасын жасау керек , тәуелді кітапханаларды орнату және Pycharm ішінде ортаны орнату .

Бұл қадамдар үшін онлайн орнату арасында таңдау жасау үшін төмендегі оқулықты орындаңыз (рір орнату буманы тікелей жүктеп алады онлайн) немесе офлайн тәуелділік пакеті (блогер ұсынған) (Офлайн пакетті тікелей орнату) Орнатудың екі әдісінің бірі:

1. Python ортасын конфигурациялау  
оқулығы: <https://deepcode.blog.csdn.net/article/details/136639396>  
( 2 немесе 3 әдістердің бірін таңдауға болады );

2. Офлайн тәуелділік бумаларына арналған орнату  
нұсқаулығы: <https://deepcode.blog.csdn.net/article/details/136650641>  
( 2 және 3 әдістердің бірін таңдауға болады );

Орнату үшін офлайн пакет әдісін пайдалансаңыз, желіден тыс тәуелділік кітапханасын жүктеп алыңыз, жүктеп алу мекенжайы:

[https://pan.baidu.com/s/1uHbU9YzSqN0YP\\_dTHBgpFw?pwd=mt8u](https://pan.baidu.com/s/1uHbU9YzSqN0YP_dTHBgpFw?pwd=mt8u)

(шығару коды: mt8u).

## ҚОРЫТЫНДЫ

Бұл - дипломды жобада дәл ауыл шаруашылығы саласында YOLOv8/v7/v6/v5 негізіндегі терең оқыту үлгілерін қолдану бойынша терең зерттеулер мен тәжірибелер жүргізді және осы жетілдірілген алгоритмдерді біріктіретін қызанақ піскенін анықтау жүйесін сәтті әзірледі. YOLO моделінің бірнеше нұсқаларын егжей-тегжейлі салыстыру және жүйені оңтайландыру арқылы бұл зерттеу қызанақтың піскендігін анықтаудың дәлдігі мен нақты уақыттағы жауап беру қабілетін жақсартып қана қоймай, сонымен қатар Streamlit Enables пайдаланушылары арқылы интуитивті, әдемі және пайдаланушыға ыңғайлы веб-бағдарлама интерфейсін жасады ауылшаруашылық өндірісінде және тамақ өңдеу өнеркәсібінде маңызды практикалық маңызы бар қызанақ піскендігін оңай бағалау.

Бірқатар қатаң тәжірибелік тексерулерден кейін осы мақалада ұсынылған жүйе қызанақтың піскендігін анықтау дәлдігі мен өңдеу жылдамдығы бойынша салалық жетекші деңгейге жетті. Сонымен қатар, бұл зерттеу деректер жиынын өңдеуден модельді оқытуға, болжауға және Streamlit негізіндегі жүйені жобалау мен енгізу мәліметтеріне дейін толық нұсқаулықты қамтамасыз етеді, болашақ зерттеушілер мен әзірлеушілерге эксперименттерді жаңғырту және одан әрі дамыту үшін ыңғайлылықты қамтамасыз етеді. Керемет нәтижелерге қарамастан, әртүрлі сорттар, жетілу деңгейлері және қоршаған орта жағдайлары бойынша қызанақты анықтауға қатысты жүйе әлі де көптеген қиындықтарға тап болады.

Болашақ жұмыс бағыттары мыналарды қамтиды:

- **Модельді оңтайландыру** : Жүйенің өнімділігі мен бейімделгіштігін одан әрі жақсарту үшін нейрондық желі архитектурасын іздеу (NAS) технологиясы сияқты тиімдірек желі құрылымдарын және оңтайландыру стратегияларын зерттеуді жалғастырыңыз.

- **Көп бұрышты анықтау** : қызанақтардың піскендігін бірнеше бұрыштан түсіріп, анықтай алатын, осылайша анықтау дәлдігін жақсартатын жетілдірілген кескін өңдеу технологиясын жасаңыз.

- **Қоршаған ортаға бейімделу** : әртүрлі өріс орталарында тұрақты жұмысты қамтамасыз ету үшін әртүрлі жарықтандыру және фондық жағдайлар кезінде үлгінің беріктігін жақсартыңыз.

- **Пайдаланушының өзара әрекеттесу тәжірибесі** : Техникалық емес пайдаланушылар операцияны оңай меңгере алатындай жүйе интерфейсін оңтайландырыңыз, пайдаланушы тәжірибесін одан әрі жақсартыңыз.

- **Қолдануды практикалық кеңейту** : Жүйені ауыл шаруашылығы өндірісінің байланыстарының кең ауқымына қолданыңыз, мысалы, отырғызуды басқару, егінді болжау және т.б., оның нақты ауыл шаруашылығы өндірісіндегі мәнін барынша арттыру.

Қорытындылай келе, YOLOv8/v7/v6/v5 негізіндегі қызанақтың жетілуін анықтау жүйесі технологиялық жетілудің жылдам жолында және қолдану сценарийлерінің үздіксіз кеңеюімен бұл жүйе smart ауыл шаруашылығында және

ақылды тағамда қолданылады деп күтілуде. тізбектер болашақта басқа салаларда маңызды рөл атқарады.



## ПАЙДАЛАНЫЛҒАН ӘДЕБИЕТТЕР ТІЗІМІ

[1] Фарук Дж, Муаз М, т.б. Шағын нысандар үшін оңтайландырылған архитектурасы бар бөгде заттардың қалдықтарын анықтауға арналған жақсартылған YOLOv8[J]. Мультимедиялық құралдар мен қолданбалар, 2023: 1-27.

[2] Carion N, Massa F, Synnaeve G және т.б. Трансформаторлар көмегімен объектінің ұшын анықтау[C]//Компьютерлік көру бойынша еуропалық конференция. Чам: Springer International Publishing, 2020: 213-229.

[3] Досовицкий А, Бейер Л, Колесников А, т.б. Кескін 16x16 сөзден тұрады: [J] масштабында кескінді тануға арналған трансформаторлар. arXiv алдын ала басып шығару arXiv:2010.11929, 2020.

[4] Liu Z, Lin Y, Cao Y және т.б. Свин трансформаторы: жылжымалы терезелерді пайдаланатын иерархиялық көру трансформаторы[C]//Компьютерлік көру бойынша IEEE/CVF халықаралық конференциясының материалдары. 2021 ж.: 10012-10022.

[5] Ву В, Лю Х, Ли Л және т.б. YOLO v5 алгоритмімен біріктірілген жергілікті толық конволюционды нейрондық желіні қашықтан зондтау кескінін шағын мақсатты анықтауда қолдану[J]. PloS one, 2021, 16(10): e0259283.

ҚАЗАҚСТАН РЕСПУБЛИКАСЫНЫҢ БІЛІМ ЖӘНЕ ҒЫЛЫМ МИНИСТРЛІГІ  
Қ.И. СӘТБАЕВ атындағы ҚАЗАҚ ҰЛТТЫҚ ТЕХНИКАЛЫҚ ЗЕРТТЕУ УНИВЕРСИТЕТІ

**6B07113 – «Робототехника және мехатроника» мамандығының студенті**

**Бақыт Арман Нұржанұлы**  
**дипломдық жобасына (жұмысына)**

## СЫН ПІКІР

Тақырыбы: **Машиналық оқыту арқылы піскен қызанақты тану жүйесі**

Әзірленген:

- а) графикалық бөлімі 18 парак
- б) түсіндірме жазбасы 40 бетте

## ЖҰМЫСҚА ЕСКЕРТУ ЖАСАУ

Бақыт Арман дипломдық жұмысының тақырыбы “Машиналық оқыту арқылы піскен қызанақты тану жүйесі”. Бұл дипломдық жобада Машиналық оқыту әдістерін қолдана отырып, піскен қызанақты тану жүйесін зерттеуге арналған.

Зерттеудің негізгі мақсаты тамақ өнімдерін шығаратын кәсіпорындарда қызанақты сұрыптау сапасын жақсарту арқылы, машиналық оқыту жүйесін пайдалану.

Бірінші тарауда машиналық оқыту арқылы піскен қызанақты тану жүйесі жайлы зерттеу әдістері, шешілетін мәселелер және оның шешімі жайлы сипатталады.

Екінші тарауда жұмыста қызанақтың пісу дәрежесін анықтау үшін кескінді өңдеуге және нейрондық желілерді пайдалануға негізделген алгоритм ұсынылады. Тәжірибелер жүйенің тиімділігін және оны егін жинау мен сұрыптау процесін автоматтандыру үшін тәжірибеде қолдану мүмкіндігін көрсетемін. Олар салыстырылып, жалпы кемшіліктер анықталады.

Соңғы тарауда зерттеудің негізгі нәтижелері машиналық жүйенің әдістері ауылшаруашылық кәсіпорындары үшін пайдалы екендігі дәлелденеді.

Дипломдық жоба ұйымдық стандарттарға сәйкес орындалды, аударма, жазу және сызбалар тұрғысынан сауатты құрастырылды. Ол мәтіндік және графикалық материалдарды жасауға, ұсынуға, безендіруге және мазмұнға қойылатын жалпы талаптарға сәйкес жасалған.

## ЖҰМЫС БАҒАСЫ

Студент Бақыт Арман “Машиналық оқыту арқылы піскен қызанақты тану жүйесі” атты жұмыстық жобасы жасалды, толығымен орындалған деп есептеймін. Дипломдық жоба 70 бағаланып, студент Бақыт Арманның бакалавр академиялық дәрежесіне лайық деп есептеймін.

**Сын пікір беруші**  
эл-Фараби атындағы ҚазҰУ,  
Жасанды интеллект және Big Data  
кафедрасының доценті, PhD  
«06» маусым 2024 ж



Карымсакова Н.Т.

**6B07111 – «Робототехника және мехатроника» мамандығының студенті**  
**Бақыт Арманның**  
**дипломдық жобасына (жұмысына)**

**ПІКІР**

Бақыт Арманның бакалаврлық дипломдық жобасы «Машиналық оқыту арқылы піскен қызанақты тану жүйесі» арналған.

Жұмыстың мақсаты көру қабілеті бұзылған адамдарға арналған сенсорлық көмекші құрылғы жасау және оның көмегі арқылы қоршаған ортаны қабылдауға бейімдеу болып табылады.

Жұмыстың мақсаты тамақ өнімдерін шығаратын кәсіпорындарда қызанақты сұрыптау сапасын жақсарту арқылы, машиналық оқыту жүйесін пайдалану.

Бірінші тарауда машиналық оқыту арқылы піскен қызанақты тану жүйесі жайлы зерттеу әдістері, шешілетін мәселелер және оның шешімі жайлы сипатталады.

Екінші тарауда жұмыста қызанақтың пісу дәрежесін анықтау үшін кескінді өңдеуге және нейрондық желілерді пайдалануға негізделген алгоритм ұсынылады. Соңғы тарауда зерттеудің негізгі нәтижелері машиналық жүйенің әдістері ауылшаруашылық кәсіпорындары үшін пайдалы екендігі дәлелденеді.

Дипломдық жобада мәтіндік және графикалық материалдардың құрылуына, баяндалуына, ресімделуіне және мазмұнына қойылатын жалпы талаптарға сәйкес ұйым стандарты бойынша жасалған. Студент Бақыт Арман Машиналық оқыту арқылы піскен қызанақты тану жүйесі атты жұмыстық жобасы өте жақсы дәрежеде жасалып, толығымен орындалған деп есептеймін. Дипломдық жоба өте жақсы бағаланып, студент Бақыт Арман бакалавр академиялық дәрежесіне лайық деп есептеймін.

**Ғылыми жетекші**

**Т.Ғ.М., аға оқытушы**

М.М. Жамуратова Жамуратова М.М.

«31» мамыр 2024ж.



## Метаданные

Название

**Машиналық оқыту арқылы піскен қызанақты тану жүйесі**

Автор

**Бақыт Арман Нұржанұлы**

Научный руководитель / Эксперт






**Махаббат Жамуратова**

Подразделение

**ИАиИТ**

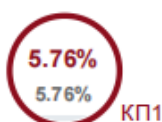
## Тревога

В этом разделе вы найдете информацию, касающуюся текстовых искажений. Эти искажения в тексте могут говорить о ВОЗМОЖНЫХ манипуляциях в тексте. Искажения в тексте могут носить преднамеренный характер, но чаще, характер технических ошибок при конвертации документа и его сохранении, поэтому мы рекомендуем вам подходить к анализу этого модуля со всей долей ответственности. В случае возникновения вопросов, просим обращаться в нашу службу поддержки.

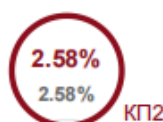
Замена букв		31
Интервалы		0
Микропробелы		107
Белые знаки		9
Парафразы (SmartMarks)		14

## Объем найденных подобиий

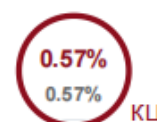
КП-ия определяют, какой процент текста по отношению к общему объему текста был найден в различных источниках.. Обратите внимание!Высокие значения коэффициентов не означают плагиат. Отчет должен быть проанализирован экспертом.

**25**

Длина фразы для коэффициента подобия 2

**6852**

Количество слов

**54721**

Количество символов

## Поиск контента ИИ

Интегрированный модуль поиска контента AI. Нажмите «Подробнее», чтобы узнать больше о результатах и алгоритме поиска.

### Коэффициент вероятности ИИ



## Подобия по списку источников

Ниже представлен список источников. В этом списке представлены источники из различных баз данных. Цвет текста означает в каком источнике он был найден. Эти источники и значения Коэффициента Подобия не отражают прямого плагиата. Необходимо открыть каждый источник и проанализировать содержание и правильность оформления источника.

### 10 самых длинных фраз

### Цвет текста

ПОРЯДКОВЫЙ НОМЕР	НАЗВАНИЕ И АДРЕС ИСТОЧНИКА URL (НАЗВАНИЕ БАЗЫ)	КОЛИЧЕСТВО ИДЕНТИЧНЫХ СЛОВ (ФРАГМЕНТОВ)	
1	<a href="https://www.cnblogs.com/deeppython/p/18097181">https://www.cnblogs.com/deeppython/p/18097181</a>	45	0.66 %
2	<a href="https://www.cnblogs.com/deeppython/p/18092916">https://www.cnblogs.com/deeppython/p/18092916</a>	37	0.54 %
3	<a href="https://www.cnblogs.com/deeppython/p/18097181">https://www.cnblogs.com/deeppython/p/18097181</a>	34	0.50 %
4	<a href="https://www.cnblogs.com/deeppython/p/18097181">https://www.cnblogs.com/deeppython/p/18097181</a>	33	0.48 %
5	<a href="https://www.cnblogs.com/deeppython/p/18097181">https://www.cnblogs.com/deeppython/p/18097181</a>	28	0.41 %
6	<a href="https://www.cnblogs.com/deeppython/p/18100262">https://www.cnblogs.com/deeppython/p/18100262</a>	23	0.34 %
7	<a href="https://www.cnblogs.com/deeppython/p/18097181">https://www.cnblogs.com/deeppython/p/18097181</a>	22	0.32 %
8	<a href="https://www.cnblogs.com/deeppython/p/18097181">https://www.cnblogs.com/deeppython/p/18097181</a>	21	0.31 %
9	<a href="https://www.cnblogs.com/deeppython/p/18097181">https://www.cnblogs.com/deeppython/p/18097181</a>	18	0.26 %
10	<a href="https://www.cnblogs.com/deeppython/p/18097181">https://www.cnblogs.com/deeppython/p/18097181</a>	16	0.23 %