11-дәріс Интеллектуалды жүйелердегі компьютерлік көру

Пәні: Интеллектуалды ақпараттық жүйелер

Алдыңғы тақырып: Табиғи тілдерді өңдеу NLP

Бұл дәріс цифрлық суреттің құрылымынан бастап CNN, нысандарды анықтау, сегментация және жауапты қолдануға дейінгі толық жолды түсіндіреді. Материал 25 слайдпен бірдей ретте құрылған, сондықтан мәтін мен презентацияны қатар пайдалануға болады.

Дәрістің мақсаты

Студенттерге суретті сандық түрде көрсету, жергілікті белгілерді свёртка арқылы алу, компьютерлік көру міндеттерін ажырату және модель сапасын дұрыс бағалау қағидаларын түсіндіру.

Күтілетін оқу нәтижелері

- пиксель, арна және сурет тензорын түсіндіреді
- свёртка, stride, padding және pooling қызметін байланыстырады
- классификация, детекция және сегментация міндеттерін ажыратады
- CNN мен transfer learning тәсілдерін түсіндіреді
- accuracy, precision, recall, IoU және mAP метрикаларын таңдайды
- құпиялық, әділдік және адам бақылауының маңызын негіздейді

Компьютерлік көру міндеттерінің картасы

Міндет	Нәтиже	Мысал	Негізгі метрика
Классификация	Бір класс	Суретте мысық бар	Accuracy және F1
Детекция	Класс және қорап	Көліктің орнын табу	IoU және mAP
Сегментация	Пиксель маскасы	Жол жолағын бөлу	IoU немесе Dice
OCR	Таңбалар мәтіні	Құжатты оқу	CER немесе WER

Компьютерлік көру ұғымы

Адам суретке қарағанда нысанды, оның орнын және жағдайын бірден байқайды. Компьютер үшін сурет бастапқыда тек сандар кестесі. Сондықтан жүйеге пиксельдерден мағыналы белгі табуды үйретеміз.

Компьютерлік көру сурет пен бейнеден ақпарат алуға арналған әдістерді біріктіреді. Ол нысанды жіктейді, суреттегі орнын табады, әр пиксельді бөледі және мәтінді таниды. Бұл міндеттер бір-біріне ұқсас болғанымен, нәтижесі мен бағалау тәсілі әртүрлі.

Қарапайым мысал. Кіру жүйесі камерадан студенттік билетті көреді, фотосуретті анықтайды және рұқсат етілген тұлғамен салыстырады.

Негізгі ой: компьютерлік көру пиксельдерден нысан, орын және мағына туралы қорытынды шығарады.

Дәрістің мақсаты және оқу нәтижелері

Енді дәрістің мақсатын нақтылап алайық.

Алдымен бүгінгі мақсатты нақтылаймын. Мен суретті сандық матрица ретінде түсіндіруден бастап, CNN, object detection және segmentation жүйелеріне дейін бірізді өтемін.

Дәріс соңында сіздер пиксель, арна, ядро, свёртка, feature map және pooling ұғымдарын түсіндіре аласыздар. Сонымен қатар классификация, детекция және сегментация міндеттерін ажыратып, accuracy, precision, recall және IoU метрикаларын дұрыс таңдайсыздар.

Қарапайым мысал. Суреттен көлік түрін анықтау тапсырмасы берілсе, сіздер оны классификация деп атайсыздар. Ал көліктің суреттегі орнын төрт координатамен көрсету қажет болса, детекция міндетін таңдайсыздар.

Негізгі ой: біз суреттен жауап алудың толық құбырын дерек, модель, бағалау және жауапты қолданумен бірге қараймыз.

Компьютерлік көру не үшін қажет

Осы түсініктен кейін практикалық қажеттілікке көшемін.

Енді компьютерлік көрудің практикалық маңызын ашамын. Камералар, медициналық құрылғылар, спутниктер және өндірістік датчиктер күн сайын үлкен көлемде визуалды дерек жасайды. Оны қолмен толық тексеру баяу және біркелкі емес болуы мүмкін.

Компьютерлік көру сапаны бақылау, құжат оқу, қауіпсіздік, навигация, медицина және ауыл шаруашылығында қолданылады. Автоматтандыру маманды алмастыру үшін емес, қайталанатын тексеруді жеңілдету және күмәнді жағдайды ертерек көрсету үшін пайдалы.

Қарапайым мысал. Өндіріс желісінде камера әр өнімді түсіреді. Жүйе қаптамадағы жыртылуды байқаса, оны қызметкерге қосымша тексеруге жібереді.

Негізгі ой: компьютерлік көру визуалды ағынды реттеп, адамға маңызды жағдайларға назар аударуға көмектеседі.

Цифрлық сурет және пиксель

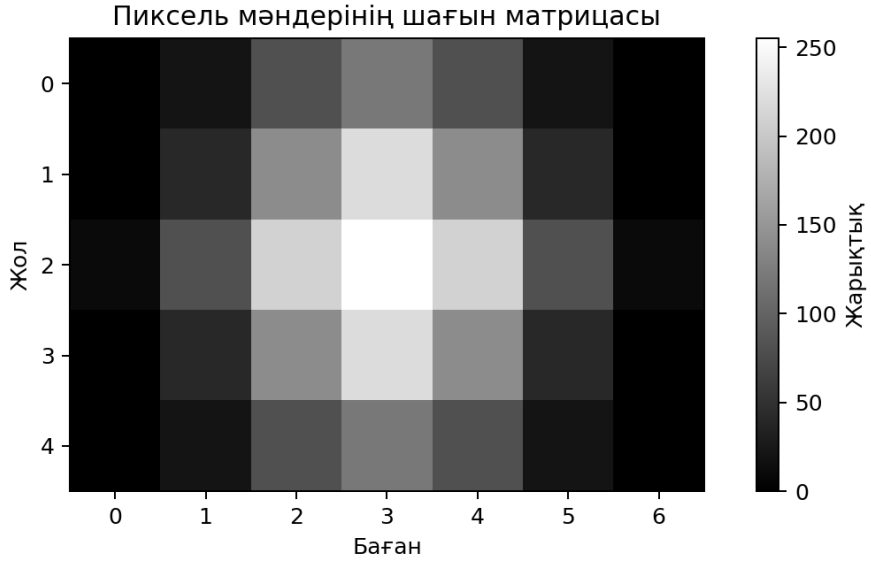
Енді суреттің компьютерде қалай сақталатынын қарастырамын.

Компьютер суретті кішкентай ұяшықтардан тұратын тор ретінде сақтайды. Әр ұяшық пиксель деп аталады. Пиксельдің мәні жарықтықты немесе түсті көрсетеді. Суретті үлкейткенде осы торды көруге болады.

Суреттің ажыратымдылығы ені мен биіктігіндегі пиксель санымен беріледі. 1920 × 1080 суретте екі миллионнан астам пиксель бар. Ажыратымдылық артса, ұсақ бөлшек көбірек сақталады, бірақ жад пен есептеу шығыны да өседі.

Қарапайым мысал. 8 × 8 ақ-қара сурет 64 саннан тұрады. 0 қара, 255 ақ, ал аралық мәндер сұр реңктерді көрсетеді.

Негізгі ой: пиксель суреттің ең кіші сандық элементі, ал ажыратымдылық бөлшек пен ресурс арасындағы теңгерімді анықтайды.



1-сурет. Пиксель матрицасының оқу мысалы

Түс арналары және сурет тензоры

Осыны түсінгеннен кейін келесі қадамға өтемін.

Түсті суретте әр пиксель бір санмен шектелмейді. RGB моделінде қызыл, жасыл және көк арналары қолданылады. Әр арна жеке матрица ретінде қарастырылады, ал үшеуі бірігіп түсті сурет құрайды.

Суреттің өлшемін H × W × C түрінде жазамыз. H биіктікті, W енді, C арна санын көрсетеді. Нейрондық желі бірнеше суретті бір batch түрінде қабылдаса, өлшемге мысалдар саны да қосылады.

Қарапайым мысал. Қызыл алма суретінде қызыл арна мәндері жоғары болуы мүмкін. Бірақ жарық өзгерсе, сол алманың пиксель мәндері де өзгереді. Сондықтан модель тек бір түске тәуелді болмауы керек.

Негізгі ой: түсті сурет бірнеше арнадан құралған үшөлшемді сандық тензор.

Суретті алдын ала өңдеу

Бұл бөлім алдыңғы тәсілдің мүмкіндігі мен шектеуін толықтырады.

Модельге сурет бермес бұрын өлшемін, мәндер ауқымын және сапасын реттеймін. Resize барлық суретті бір пішінге келтіреді, normalization пиксель мәндерін ыңғайлы диапазонға ауыстырады, ал кей жағдайда шуды азайтамыз.

Өңдеу кезінде пропорцияны бұзбау маңызды. Тікбұрышты нысанды квадратқа күшпен созсақ, пішін өзгеріп кетеді. Сондықтан crop, padding немесе пропорцияны сақтайтын resize тәсілін таңдаймыз.

Қарапайым мысал. 3000 × 2000 суретті 224 × 224 өлшеміне келтіргенде модель жылдам жұмыс істейді, бірақ ұсақ мәтін жоғалуы мүмкін. OCR үшін жоғарырақ ажыратымдылық қажет болуы ықтимал.

Негізгі ой: алдын ала өңдеу суреттерді біріздендіреді, бірақ маңызды бөлшекті жоғалтпау керек.

Сүзгі және свёртка идеясы

Енді сұрақты сәл тереңдетіп қарастырамын.

Енді суреттен белгі қалай алынатынын түсіндіремін. Кішкентай сандар матрицасын ядро немесе сүзгі деп атаймыз. Ол сурет бойымен қозғалып, әр аймақта пиксельдермен көбейту және қосу амалын орындайды.

Ядро жиек, бұрыш немесе текстура сияқты жергілікті үлгіні күшейте алады. Бір сүзгі бір белгіні іздейді, ал бірнеше сүзгі әртүрлі белгілер картасын жасайды. Бұл амал convolution немесе свёртка деп аталады.

Қарапайым мысал. Қарадан аққа күрт ауысқан жерде жиек сүзгісінің жауабы үлкен болады. Біртүсті аймақта жауап кіші болады.

Негізгі ой: свёртка кішкентай сүзгіні сурет бойымен жүргізіп, жергілікті белгілер картасын құрады.

Аудиторияға сұрақ: бір сүзгі суреттің әр бөлігінде неге бірдей белгі іздей алады?

Свёртка формуласы stride және padding

Енді алдыңғы ойды келесі ұғыммен байланыстырамын.

Свёртка нәтижесінің өлшемі ядроға, қадамға және шет толтыруға байланысты. Stride сүзгінің қанша пиксельге жылжитынын, padding сурет шетіне қосылатын мәндерді көрсетеді.

Бір өлшем үшін шығыс өлшемі: O = floor((N + 2P - K) / S) + 1. Мұнда N кіріс өлшемі, K ядро өлшемі, P padding, S stride. Қадам үлкейсе, карта кішірейеді. Padding қолданылса, шеттегі ақпарат жақсырақ сақталады.

Қарапайым мысал. N=32, K=3, P=1, S=1 болса, шығыс өлшемі 32 болады. Егер S=2 болса, өлшем шамамен екі есе азаяды.

Негізгі ой: stride есептеу тығыздығын, padding шеттегі ақпараттың сақталуын басқарады.

O = floor((N + 2P - K) / S) + 1

Жиектерді анықтау

Осыны түсінгеннен кейін келесі қадамға өтемін.

Жиек суреттегі жарықтықтың күрт өзгерген орнын көрсетеді. Нысан шекарасы, мәтін сызығы және пішіннің маңызды бөлігі жиек арқылы көрінеді. Классикалық әдістер Sobel немесе Canny сүзгілерін қолданады.

Жиек картасы суретті толық түсіндірмейді, бірақ пішін туралы ықшам ақпарат береді. Күшті шу жалған жиектер жасай алады, сондықтан алдын ала тегістеу пайдалы болуы мүмкін.

Қарапайым мысал. Ақ қағаздағы қара әріптердің шекарасы айқын. OCR жүйесі алдымен осы контурларды пайдаланып таңбаларды бөлуі мүмкін.

Негізгі ой: жиек жарықтық өзгерісін көрсетеді және нысан пішінін табуға көмектеседі.

Классикалық компьютерлік көру құбыры

Бұл бөлім алдыңғы тәсілдің мүмкіндігі мен шектеуін толықтырады.

Терең оқытуға дейін маман пайдалы белгілерді қолмен жобалады. Құбыр суретті өңдеу, жиек немесе бұрыш табу, дескриптор жасау және классификаторға беру кезеңдерінен тұрды.

HOG пішін бағыттарын, SIFT жергілікті тұрақты нүктелерді сипаттайды. Бұл әдістер аз дерек пен түсіндірілетін белгілер қажет болғанда әлі де пайдалы. Бірақ күрделі ортада қолмен таңдалған белгі жеткіліксіз болуы мүмкін.

Қарапайым мысал. Құжаттағы мөрді табу үшін түс диапазоны, дөңгелек пішін және жиек белгілерін біріктіруге болады.

Негізгі ой: классикалық тәсілде белгіні адам жобалайды, ал модель дайын белгілер бойынша шешім шығарады.

CNN архитектурасы

Енді сұрақты сәл тереңдетіп қарастырамын.

CNN суретке бейімделген нейрондық желі. Ол свёртка қабаттары арқылы жергілікті белгілерді үйренеді, активация сызықтық емес байланыс қосады, pooling өлшемді кішірейтеді, ал соңғы қабат нәтиже береді.

Алғашқы қабаттар жиек пен қарапайым текстураны, ортаңғы қабаттар бөлшектерді, жоғарғы қабаттар тұтас нысанға ұқсас көріністі үйренуі мүмкін. Салмақтарды ортақ пайдалану параметр санын азайтады.

Қарапайым мысал. Мысық суретін танитын желі алдымен жиектерді, кейін құлақ пен көз пішінін, соңында тұтас бейнені біріктіреді.

Негізгі ой: CNN қарапайым жергілікті белгілерден күрделі визуалды көрініс құрады.

Аудиторияға сұрақ: CNN алғашқы және соңғы қабаттарында қандай белгілерді үйренуі мүмкін?

Feature map және рецептивті өріс

Енді алдыңғы ойды келесі ұғыммен байланыстырамын.

Әр свёртка сүзгісі feature map деп аталатын белгі картасын жасайды. Картадағы жоғары мән сүзгі іздеген үлгінің сол аймақта бар екенін көрсетеді.

Нейронның рецептивті өрісі кіріс суреттің оған әсер ететін бөлігін білдіреді. Қабаттар тереңдеген сайын рецептивті өріс ұлғайып, модель үлкенірек контексті көреді.

Қарапайым мысал. Алғашқы қабаттағы нейрон кішкентай жиекті көреді. Кейінгі қабаттағы нейрон бірнеше жиекті біріктіріп, есік немесе терезе бөлігіне жауап беруі мүмкін.

Негізгі ой: feature map белгінің қай жерде табылғанын, рецептивті өріс модельдің қанша контекст көретінін көрсетеді.

Pooling және өлшемді азайту

Осыны түсінгеннен кейін келесі қадамға өтемін.

Pooling белгі картасының өлшемін азайтады. Max pooling шағын терезедегі ең үлкен мәнді, average pooling орташа мәнді алады. Бұл есептеуді жеңілдетіп, ұсақ ығысуларға тұрақтылық беруі мүмкін.

Өлшемді тым ерте немесе тым қатты азайтсақ, ұсақ нысандар жоғалады. Сондықтан pooling саны мен орны міндетке байланысты таңдалады. Кейбір заманауи архитектуралар өлшемді stride арқылы азайтады.

Қарапайым мысал. 2 × 2 терезедегі [1, 5; 2, 3] мәндерінің max pooling нәтижесі 5 болады.

Негізгі ой: pooling маңызды жауапты сақтап, кеңістіктік өлшемді азайтады, бірақ ұсақ бөлшекті жоғалтуы мүмкін.

Деректерді көбейту және оқыту

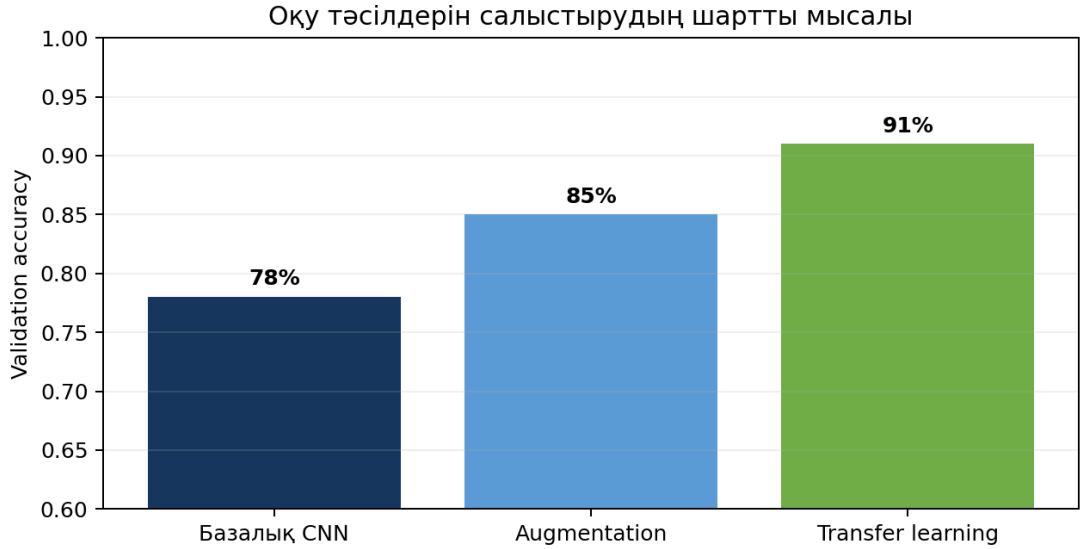
Бұл бөлім алдыңғы тәсілдің мүмкіндігі мен шектеуін толықтырады.

Көру моделі әртүрлі жарық, бұрыш және фон жағдайында жұмыс істеуі керек. Data augmentation оқыту суретін бұру, шағылыстыру, crop жасау немесе жарықты өзгерту арқылы жаңа нұсқалар жасайды.

Augmentation шынайы қолдану жағдайына ұқсас болуы керек. Медициналық суретті мағынасы өзгеретіндей аудару немесе цифрды бұру қате белгі жасауы мүмкін. Оқыту, валидация және тест деректері бір нысанның көшірмелерімен араласпауы керек.

Қарапайым мысал. Мың өнім суретін әртүрлі жарық және шағын бұрылыс арқылы көбейтіп, камера жағдайының өзгеруіне төзімді модель үйретуге болады.

Негізгі ой: деректерді көбейту әртүрлілікті арттырады, бірақ нысанның мағынасын сақтауы тиіс.



2-сурет. Оқыту тәсілдерін салыстырудың шартты графигі

Суретті классификациялау

Енді сұрақты сәл тереңдетіп қарастырамын.

Классификация бүкіл суретке бір немесе бірнеше класс береді. Модель соңғы қабатта әр класс үшін логит шығарып, softmax арқылы ықтималдыққа айналдырады.

Формула: $P(y=i) = \exp(z_i) / \Sigma \exp(z_j)$. Ең үлкен ықтималдық таңдалуы мүмкін, бірақ сенімділік шегі мен қате салдары да ескеріледі. Бір суретте бірнеше нысан болса, multi-label классификация қажет.

Қарапайым мысал. Жүйе жапырақ суретін «сау», «саңырауқұлақ ауруы» немесе «зиянкес белгісі» деп бөледі.

Негізгі ой: классификация суреттің жалпы класын болжайды, бірақ нысанның орнын көрсетпейді.

Нысандарды анықтау және IoU

Енді алдыңғы ойды келесі ұғыммен байланыстырамын.

Object detection суреттегі нысан класын және орнын бір уақытта табады. Орны bounding box арқылы беріледі: x, y, ені және биіктігі. Бір суретте бірнеше қорап болуы мүмкін.

Болжам қорабы мен нақты қораптың қабысуын IoU өлшейді: IoU = Intersection / Union. 1 толық сәйкесуді, 0 қабысудың жоқтығын көрсетеді. Белгілі бір шектен жоғары IoU дұрыс табу ретінде есептеледі.

Қарапайым мысал. Көше суретінде модель әр көліктің айналасына қорап салып, «жеңіл көлік», «автобус» немесе «жүк көлігі» белгісін береді.

Негізгі ой: детекция нысанның не екенін және суреттің қай жерінде тұрғанын бірге анықтайды.

Аудиторияға сұрақ: IoU мәні жоғары болуы үшін қандай екі аймақ қабысуы керек?

Сегментация

Осыны түсінгеннен кейін келесі қадамға өтемін.

Сегментация әр пиксельге класс береді. Semantic segmentation бір класқа жататын барлық пиксельді біріктіреді. Instance segmentation бір кластағы жеке нысандарды бөлек көрсетеді.

Бұл міндет bounding box-қа қарағанда дәлірек шекара береді, бірақ белгілеу қымбат. Медициналық кескін, жол жолағы және жер бедерін талдауда пиксельдік шекара маңызды болуы мүмкін.

Қарапайым мысал. Суреттегі үш алмаға semantic segmentation бір «алма» маскасын береді, ал instance segmentation әр алманы жеке маскамен белгілейді.

Негізгі ой: сегментация нысан шекарасын пиксель деңгейінде анықтайды.

OCR және құжаттарды тану

Бұл бөлім алдыңғы тәсілдің мүмкіндігі мен шектеуін толықтырады.

OCR суреттегі мәтінді машиналық оқылатын таңбаларға айналдырады. Құбыр құжатты түзету, мәтін аймағын табу, жолдарға бөлу, таңбаларды тану және тілдік тексеруден тұруы мүмкін.

Сапа қаріпке, жарыққа, бұрылуға және сурет анықтылығына тәуелді. Кестелер мен мөрлер құрылымды күрделендіреді. Құжат дерегінде құпиялық пен қате түзету мүмкіндігі маңызды.

Қарапайым мысал. Студент анықтамасының суретінен аты-жөні, күні және нөмірі оқылып, ақпараттық жүйедегі өрістерге енгізіледі.

Негізгі ой: OCR пиксельдегі мәтінді таңбаға айналдырып, құжатты құрылымдауға мүмкіндік береді.

Бетті тану және эмбеддинг

Енді сұрақты сәл тереңдетіп қарастырамын.

Бетті тану алдымен бетті табады, кейін оның ерекшелігін векторға айналдырады. Екі эмбеддинг арасындағы қашықтық немесе косинустық ұқсастық тұлғалардың сәйкестігін бағалайды.

Жүйе жарық, бұрыш, жас өзгерісі және камера сапасына сезімтал. Бір суретпен шешім шығару қауіпті болуы мүмкін. Жалған сәйкестік пен жалған бас тартудың салдары бөлек бағаланады.

Қарапайым мысал. Телефонды ашу кезінде жаңа бет эмбеддингі тіркелген эталонмен салыстырылады. Ұқсастық шектен асса, қосымша қорғаныс шарттарымен рұқсат беріледі.

Негізгі ой: бетті тану суретті емес, тұлғаның ықшам векторлық көріністерін салыстырады.

Transfer learning

Енді алдыңғы ойды келесі ұғыммен байланыстырамын.

Үлкен сурет жинағында алдын ала оқытылған CNN жалпы жиек, текстура және пішін белгілерін меңгереді. Жаңа міндетте осы салмақтарды бастапқы нүкте ретінде қолданамыз.

Feature extraction кезінде негізгі қабаттарды бекітіп, тек соңғы классификаторды оқытамыз. Fine tuning кезінде жоғарғы қабаттардың бір бөлігін шағын оқу жылдамдығымен жаңартамыз. Жаңа сала бастапқы деректен қатты ерекшеленсе, көбірек бейімдеу қажет.

Қарапайым мысал. Жалпы суреттерде оқытылған модельді 800 өсімдік жапырағы суретімен ауруды тануға бейімдеуге болады.

Негізгі ой: transfer learning аз дерекпен оқытуды жылдамдатады, бірақ бастапқы модельдің ығысуын да бірге әкелуі мүмкін.

Аудиторияға сұрақ: transfer learning қай кезде нөлден оқытудан тиімді?

Классификация метрикалары

Осыны түсінгеннен кейін келесі қадамға өтемін.

Модель сапасын тест жиынында бағалаймын. Accuracy барлық дұрыс болжам үлесін береді. Precision оң деп белгіленгендердің қаншасы дұрыс екенін, recall нақты оң мысалдардың қаншасы табылғанын көрсетеді.

F1 precision мен recall теңгерімін көрсетеді. Кластар тең болмаса, әр класс бойынша метриканы бөлек қараймыз. Қате салдары жоғары жағдайда сенімділік интервалы мен адам тексеруі де қажет.

Қарапайым мысал. Ақаулы өнім 1 пайыз болса, бәрін «қалыпты» дейтін модель 99 пайыз accuracy алады, бірақ ақаудың recall мәні нөл болады.

Негізгі ой: метрика дерек теңгерімі мен қате салдарына сай таңдалады.

Детекция метрикалары IoU және mAP

Бұл бөлім алдыңғы тәсілдің мүмкіндігі мен шектеуін толықтырады.

Детекцияда классификация дұрыстығына қоса қораптың орналасуын бағалаймыз. IoU нақты және болжам аймақтарының қабысуын көрсетеді. Precision мен recall әр сенімділік шегінде өзгереді.

Average Precision precision-recall қисығы астындағы сапаны бір класс үшін жинақтайды. mAP бірнеше кластың AP орташа мәнін береді. IoU шегі қаталдаған сайын дәл локализация талап етіледі.

Қарапайым мысал. Бір модель үлкен көліктерді жақсы, ал алыстағы шағын көліктерді нашар табуы мүмкін. Сондықтан mAP-пен бірге нысан өлшемдері бойынша сапаны қараймыз.

Негізгі ой: mAP класс пен локализация сапасын біріктіреді, бірақ қате мысалдарды оқу бәрібір қажет.

Аудиторияға сұрақ: шағын нысандарды нашар табатын модельді бір ғана mAP саны толық сипаттай ма?

Қателерді талдау және түсіндірмелілік

Енді сұрақты сәл тереңдетіп қарастырамын.

Орташа метрика модельдің неге қателескенін айтпайды. Мен жалған оң, жалған теріс және дұрыс мысалдарды жарық, фон, бұрылыс, нысан өлшемі мен камера түрі бойынша топтаймын.

Grad-CAM сияқты әдістер модель назар аударған аймақты жуық түрде көрсетеді. Мұндай карта пайдалы белгі береді, бірақ шешімнің толық себебі немесе дұрыстық кепілі емес.

Қарапайым мысал. Модель қасқырды қарлы фон арқылы таныса, қар жоқ суретте қателесуі мүмкін. Жылу картасы модельдің нысаннан гөрі фонға қарағанын байқатады.

Негізгі ой: қате мысалдарын және назар картасын бірге талдау модельдің әлсіз жерін табуға көмектеседі.

Қолдану салалары және жүйелік талаптар

Енді алдыңғы ойды келесі ұғыммен байланыстырамын.

Компьютерлік көру медицинада, өндірісте, көлікте, ауыл шаруашылығында, білімде және құжат айналымында қолданылады. Бірақ бір архитектура барлық салаға бірдей жарамайды.

Нақты жүйеде дәлдікпен бірге жылдамдық, жад, энергия, камера сапасы және кідіріс бағаланады. Edge құрылғыда шағын модель қажет болуы мүмкін, ал серверде күрделі модель қолдануға болады.

Қарапайым мысал. Дрон егістікті түсіріп, әлсіз өсімдіктер аймағын белгілейді. Агроном карта бойынша нақты жерді тексереді және шешім қабылдайды.

Негізгі ой: модель таңдау міндетке, ресурсқа және қате салдарына тәуелді.

Жауапты қолдану және қорытынды

Осыны түсінгеннен кейін келесі қадамға өтемін.

Соңғы бөлімде техникалық сапаны жауапты қолданумен байланыстырамын. Камера адам бейнесін, орналасуын және жеке кеңістігін жазуы мүмкін. Сондықтан мақсат, сақтау мерзімі және қолжетімділік алдын ала анықталуы керек.

Модель әртүрлі тері реңкі, жас, киім немесе камера жағдайында бірдей жұмыс істемеуі мүмкін. Сапаны топтар бойынша тексеріп, маңызды шешімде адам бақылауын, шағымдану және түзету мүмкіндігін қамтамасыз етеміз.

Қарапайым мысал. Бет тану жүйесі студентті қате таныса, есік толық жабылып қалмауы керек. Балама тексеру және қызметкерге жүгіну жолы болуы тиіс.

Қорытынды ой: компьютерлік көру пиксельді белгілерге, белгілерді болжамға айналдырады. Сенімді жүйе дәлдікті дерек сапасымен, құпиялықпен және адам бақылауымен бірге қамтамасыз етеді.

Салыстыру кестелері

1-кесте. Компьютерлік көру міндеттері

Міндет	Шығыс	Белгілеу қиындығы	Мысал
Классификация	Класс	Төмен	Сурет түрі
Детекция	Класс және қорап	Орташа	Көлік орны
Сегментация	Пиксель маскасы	Жоғары	Ісік шекарасы

2-кесте. Белгілерді алу тәсілдері

Тәсіл	Белгіні кім жасайды	Артықшылығы	Шектеуі
HOG немесе SIFT	Маман	Аз деректе пайдалы	Күрделі ортаға шектеулі
CNN	Модель	Белгіні деректен үйренеді	Көп дерек пен ресурс қажет
Transfer learning	Алдын ала модель және жаңа дерек	Аз дерекпен жылдам	Бастапқы ығысу сақталуы мүмкін

3-кесте. Негізгі метрикалар

Метрика	Нені өлшейді	Қай кезде маңызды
Accuracy	Жалпы дұрыс үлес	Кластар тең болғанда
Recall	Нақты оңның табылған үлесі	Қауіпті өткізбеу керек болғанда
IoU	Аймақтардың қабысуы	Детекция мен сегментацияда
mAP	Кластар бойынша детекция сапасы	Бірнеше класс пен шек болғанда

Шағын практикалық тапсырма

Тапсырма: университет кіреберісіндегі камера жүйесін жобалаңыз. 1) жүйенің мақсатын анықтаңыз; 2) классификация, детекция немесе бет эмбеддингін қай жерде қолданатыныңызды түсіндіріңіз; 3) екі метрика таңдаңыз; 4) құпиялық пен қате шешімге арналған екі қорғаныс шарасын ұсыныңыз.

Күтілетін талдау: алдымен адамды табу үшін детекция, кейін рұқсатты тексеру үшін эмбеддинг салыстыруы қолданылуы мүмкін. False reject студентті өткізбеуі, false accept бөгде адамды кіргізуі мүмкін. Сондықтан балама тексеру мен адам бақылауы қажет.

Бақылау сұрақтары

- Пиксель мен арна деген не?
- Сурет тензорының H W C өлшемдері нені көрсетеді?
- Свёртка ядросы қалай қозғалады?
- Stride пен padding шығыс өлшеміне қалай әсер етеді?
- CNN классикалық белгілерден несімен ерекшеленеді?
- Классификация мен детекция айырмасы қандай?
- Semantic және instance segmentation қалай ерекшеленеді?
- IoU нені өлшейді?
- Тең емес кластарда accuracy неге жеткіліксіз?
- Компьютерлік көруде адам бақылауы қай кезде міндетті?

Негізгі терминдер

Термин	Қысқаша анықтама
Пиксель	Суреттің ең кіші сандық элементі
Арна	Түс немесе белгі туралы жеке матрица
Свёртка	Сүзгіні сурет бойымен жүргізіп белгі алу амалы
Feature map	Сүзгінің жауап картасы
Pooling	Белгі картасының өлшемін азайту
CNN	Свёртка қабаттарына негізделген нейрондық желі
Bounding box	Нысанды қоршаған тікбұрыш
IoU	Екі аймақ қабысуының олардың бірлескен ауданына қатынасы
Сегментация	Әр пиксельге класс беру
Transfer learning	Алдын ала оқытылған модельді жаңа міндетке бейімдеу

Ұсынылатын әдебиеттер

- Szeliski R. Computer Vision Algorithms and Applications.
- Goodfellow I., Bengio Y., Courville A. Deep Learning.
- LeCun Y., Bengio Y., Hinton G. Deep Learning.
- He K. және басқалар. Deep Residual Learning for Image Recognition.
- Redmon J. және басқалар. You Only Look Once Unified Real Time Object Detection.