

ДИФФЕРЕНЦИАЛДЫҚ ТЕНДЕУЛЕРДІ МАТЧНСАТ ОРТАСЫНДА ШЫҒАРУ ЖОЛДАРЫ

Мекебаев Нурбапа Отанович, PhD, әл-Фараби атындағы Қазақ ұлттық университеті, Алматы қ., Қазақстан, nurbara@mail.ru

Алпысбай Аружан Алпысбайқызы, магистрант, Қазақ ұлттық қыздар педагогикалық университеті

СПОСОБЫ РЕШЕНИЯ ДИФФЕРЕНЦИАЛЬНЫХ УРАВНЕНИЙ В МАТЧНСАТ

Мекебаев Нурбапа Отанович, PhD, Казахский национальный университет им. аль-Фараби, г. Алматы, Казахстан, nurbara@mail.ru

Алпысбай Аружан Алпысбайқызы, магистрант, Казахский национальный женский педагогический университет.

Аннотация. В статье студентам предлагается решение дифференциальных уравнений с помощью компьютерных программ. Одна из основных областей применения информационных технологий - математические и научно-технические расчеты. Сегодня такие расчеты очень важны. Одной из наиболее мощных и эффективных математических систем для этого является Mathcad. Mathcad-Matlab, система, которая занимает особое место среди многих подобных систем. Mathcad остается единственной системой, которая описывает решение типичных математических задач. Mathcad - это система с очень эффективным ориентированно-математическим интерфейсом и отличным инструментом для научной графики, а также позволяет выполнять численные и аналитические вычисления. В статье рассматриваются способы решения дифференциальных уравнений в MatchCat. Решение простых дифференциальных уравнений широко используется в практике научных и технических расчетов. Хотя линейные простые дифференциальные уравнения имеют решения в виде специальных функций, многие физические системы характеризуются нелинейными и линейными простыми дифференциальными уравнениями без аналитических решений. В этом случае вы должны использовать численные методы для решения простых дифференциальных уравнений.

Ключевые слова: дифференциальные уравнения, MatchCat, уравнения высшего порядка, информационные технологии, мультимедийные технологии.

The Bulletin of Kazakh Academy of Transport and Communications named after M. Tynyshpayev
ISSN 1609-1817. Vol. 116, No.1 (2021), pp.286-293

REAL-TIME OBJECT RECOGNITION BASED ON DEEP LEARNING

Abu Kuandykov, Doctor of Technical Sciences, Professor, International Kazakh-Turkish university named after Khoja Ahmed Yasawi, Turkestan, Kazakhstan, abu.kuandykov@mail.ru

Yerkebulan Taspolat, master's degree student, International Kazakh-Turkish university named after Khoja Ahmed Yasawi, Turkestan, Kazakhstan, yerkebulantaspolat@gmail.com

Abstract: In the article increasing the need to use a convolutional neural network (CNN) mobile devices with limited computing power and memory resources a number of effective architectures have been developed that address the issues of increasing research on effective modeling.

Among them, the MobileNet, ShuffleNet, and MobileNetV2 architectures proposed in recent years have been studied. Research has shown that all of these models are based on deep isolation, which is not effectively implemented in many deep learning systems.

We estimate the actual output speed of the system on specific devices. The speed is calculated based on the average processing time of 100 images by the testing tool. This time includes the pre-processing time of the image, but does not include the post-processing time of the part (decoding limit boxes and performing non-maximum clicks). Usually, post-processing is performed on a processor that can work asynchronously with other parts running on a mobile GPU.

Therefore, in this study, we propose a system that is effective for constant convolution. Then we offer a real-time object detection system by integrating the system with Single Shot MultiBox Detector (SSD). Provides our object detection and optimization system for SSD. The main goal of our optimization is to increase the speed with acceptable accuracy. With the exception of the efficient feature acquisition Network presented in the last section, we have built an object detection network in a different way than

the original SSD, carefully processing it with five function scale maps. In addition, for each opportunity map used for detection, we create a residual block before making a forecast.

Convolution, separated from depth, is not the only way to build an effective model. Instead of using depth-separated convolution, our proposed system was developed using conventional Convolution and achieved reliable results in ILSVRC 2012, VOC 2007. By combining efficient architectural design with a mobile graphics processor and hardware-optimized runtime libraries, we can implement real-time image classification and object detection forecasts on mobile devices. For example, our proposed object detection system allows high-precision detection of 23.6 FPS on Android 9 and 125 FPS on NVIDIA TX2.

Keywords: convolutional neural network, deep learning, object identification, effective modeling, ablation.

ӘОЖ 004.852

10.52167/1609-1817-2020-116-1-286-293

А.А. Қуандықов¹, Е.А. Тасполат¹

¹Қожа Ахмет Яссауи атындағы Халықаралық қазақ-түрік университеті, Түркістан қ., Қазақстан

ТЕРЕҢ ОҚЫТУ НЕГІЗІНДЕ НАҚТЫ УАҚЫТТА ОБЪЕКТІЛЕРДІ ТАҢУ

Аңдатпа. Мақалада конволюциялық нейрондық желіні (CNN) пайдалану қажеттілігінің өсуі есептеу қуаты шектеулі және жады ресурстарымен мобильді құрылғылар тиімді модельдеу бойынша зерттеулерді арттыру мәселелері қарастырылған бірқатар тиімді архитектуралар жасалды.

Олардың ішінде соңғы жылдары ұсынылған, MobileNet, ShuffleNet және MobileNetV2 архитектуралары зерттелген. Зерттеу нәтижесінде бұл модельдердің барлығы тереңдетілген оқшаулауға негізделеді, бұл көптеген терең оқу жүйелерінде тиімді іске асырылмайды деген қорытынды жасалған.

Сондықтан бұл зерттеуде біз тұрақты конволюцияға тиімді жүйе ұсынылған. Содан кейін біз жүйені Single Shot MultiBox Detector (SSD) көмегімен біріктіру арқылы нақты уақытта объектілерді анықтау жүйесін ұсынамыз.

Түйінді сөздер: конволюциялық нейрондық желі, терең оқыту, объектілерді анықтау, тиімді модельдеу, абляция.

Кіріспе

Жедел есте сақтау қабілеті бар және бюджеттік шектеулерді есептейтін CNN сапалы модельдерін басқаруға қызығушылық артып келеді. Соңғы жылдары MobileNets [1], ShuffleNet [2], NASNet-A [3], MobileNetV2 [4] сияқты көптеген инновациялық архитектуралар ұсынылды. Алайда, бұл архитектуралардың барлығы терең оқу жүесіне бөлінетін конволюцияға сүйенеді [5], оның тиімді жүзеге асқан жүйесі жоқ. Сонымен қатар, тиімді модельдерді алгоритмдермен объектілерді жылдам анықтау алғышарттарымен біріктіретін зерттеулер аз [6]. Бұл зерттеулер суреттерді классификациялау тапсырмалары үшін де, объектілерді анықтау үшін де

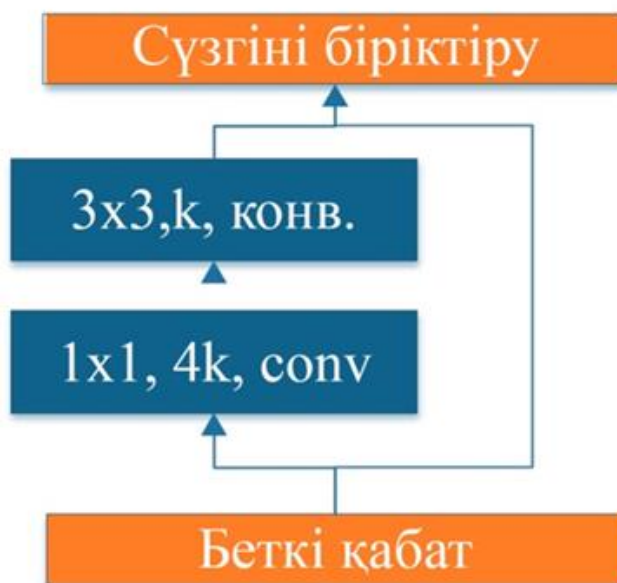
тиімді CNN архитектурасының дизайнын зерттеуге тырысады. Бұл зерттеулер төменде келтірілген бірқатар маңызды мәселелерге үлестерін қосты.

Біз DenseNet архитектурасының [7] мобильді құрылғыларға арналған нұсқасын ұсынамыз. Бұл жүйе DenseNet-ті қосу схемасын және кейбір негізгі жобалау принциптерін басшылыққа алады. Ол сондай-ақ қатаң жады мен бюджеттік шектеулерді ескере отырып жасалған. Stanford Dogs [8] деректер жиынтығындағы эксперименттік нәтижелер көрсеткендей, біздің ұсынған жүйенің түпнұсқа DenseNet архитектурасы арқылы салынған желіге қарағанда 5,05% -ға және MobileNet-тен [1] 6,53% -ға жоғары дәлдікке ие. Жүйеміз сонымен

қатар ImageNet ILSVRC 2012-де сенімді нәтижелерге қол жеткізеді [9]. Жүйеміздің бірінші дәлдігі 72,1% құрайды, бұл MobileNet-тен 1,6% жоғары. Бұл жүйе MobileNet моделінің мөлшерінің 66%-ы ғана екенін ескеру қажет. Жүйенің кейбір негізгі ерекшеліктері:

Екі жақты тығыз қабат GoogLeNet [5] жүйесі секілді біз рецептивті өрістердің

әртүрлі масштабтарын алу үшін екі жақты тығыз қабатты қолданамыз. Бір қабатты әдіс 3×3 өлшемді ядро өлшемін қолданады. Басқа деңгей әдісі үлкен нысандарға арналған визуалды үлгілерді зерттеу үшін 3×3 екі деңгейлі конволюцияны қолданады. Құрылым 1-суретте көрсетілген.



1 - сурет. Бастапқы тығыз қабат
Figure 1 – Initial dense layer

Бастапқы блогы Inception-v4 [10] және DSOD [11] негізінде біз бірінші тығыз қабаттың алдында үнемді өзек блогын құрастырамыз. Өзек блогының құрылымы 2-ші суретте көрсетілген. Бұл негізгі блок есептеу шығындарын жоғарылатпай-ақ функцияларды көрсету қабілетін тиімді түрде жақсарта алады - бірінші конволюциялық қабаттың арналарын көбейту немесе өсу қарқынын арттыру сияқты басқа әдістерге қарағанда жақсы.

Тығырық қабатындағы арналардың динамикалық саны тағы бір маңызды мәселе, тар түтік деңгейіндегі арналар саны бастапқы DenseNet-те қолданылған 4 есе өсудің орнына, кіріс формасына байланысты өзгеріп отырады. DenseNet-те алғашқы бірнеше тығыз қабаттар үшін тар арналардың саны оның кіретін арналар

санынан әлдеқайда көп екенін байқаймыз, демек бұл қабаттар үшін тар қабат өзіндік құнын төмендетудің орнына есептеу шығындарын көбейтеді. Архитектураны біртектес етіп сақтау үшін біз барлық тығыз қабаттарға тар жол қабатын қосамыз, бірақ олардың саны арнаның саны кіріс арналарының санынан аспайтын етіп енгізу формасына сәйкес динамикалық түрде реттеледі. Бастапқы DenseNet құрылымымен салыстырғанда, біздің тәжірибелеріміз бұл әдіс дәлдікке аз әсер ете отырып, есептеу шығындарын 28,5% дейін үнемдейтіндігін көрсетті. Қысымсыз өтпелі қабат. Біздің тәжірибелеріміз DenseNet ұсынған сығымдау коэффициенті функция өрнегін нашарлататынын көрсетеді. Біз әрқашан шығу арналарының санын өтпелі қабаттардағы кіріс арналарының санымен бірдей ұстаймыз.

Композициялық функция нақты жылдамдықты жақсарту үшін біз композициялық функция ретінде әдеттегі «пост-активация» данасын қолданамыз (Конволюция – пакеттік қалыпқа келтіру [12]). DenseNet-те қолданылатын алдынала белсендірудің орнына қабаттарды қорытындылау сатысында конволюциялық қабатпен біріктіруге болады. Бұл өзгерістің дәлдікке кері әсерін өтеу үшін біз таяз және кең желілік құрылымды қолданамыз. Сондай-ақ, соңғысынан кейін 1×1 конволюциялық қабатты қосамыз.

Зерттеу нәтижелері мен оларды талқылау.

Абляцияны зерттеу үшін арнайы Stanford Dogs деректер базасын құрамыз. Stanford Dogs деректер жиынтығында [8] әлемдегі 120 ит тұқымының суреттері бар. Бұл мәліметтер жиынтығы кескінді егжей-тегжейлі жіктеу үшін ImageNet-тен алынған суреттер мен аннотациялар көмегімен жасалған. Мұндай тапсырма үшін пайдаланылатын мәліметтер жиынтығы желілік архитектураның өнімділігін бағалау үшін жеткілікті екенін анықтаймыз. Алайда Stanford Dogs түпнұсқалық деректер қорында тек 14 580 жаттығу суреттері бар, олардың бір сыныпта шамамен 120 суреті бар, бұл модельді нөлден үйрету үшін жеткіліксіз. Түпнұсқа Stanford Dogs-ті пайдаланудың

орнына біз ILSVRC 2012 ішкі жиынын Stanford Dogs қолданған ImageNet желісіне сәйкес құрамыз. Оқу деректері де, растау деректері де ILSVRC 2012 мәліметтер жиынтығынан дәл көшіріледі. Осы мәліметтер жиынтығының мазмұны:

- санаттар саны – 120;
- оқылып жатқан суреттердің саны – 150 466;
- тексерілген суреттер саны – 6000.

Бүкіл желі негізгі блоктан және функционалды шығарғыштың төрт сатысынан тұрады (1-кесте). Соңғы сатыдан басқа, әр сатыдағы соңғы қабат 2-қадамның ортаңғы қабаты болып табылады. Төрт сатылы құрылым - үлкен модельдерді жобалау кезінде жиі қолданылатын құрылым. ShuffleNet [2] үш сатылы құрылымды қолданады және әр кезеңнің басында мүмкіндіктер картасының көлемін кішірейтеді. Бұл есептеу шығындарын тиімді түрде төмендетуі мүмкін болса да, ерте сатыдағы функциялар көру проблемалары үшін өте маңызды, ал функционалдық картаның уақытынан бұрын қысқарту өкілдік өнімділігін төмендетуі мүмкін деп айтамыз. Сондықтан біз төрт деңгейлі құрылымды әлі де сақтаймыз. Алғашқы екі сатыдағы қабаттар саны қолайлы диапазонға арнайы реттелген.

Кесте 1 – Архитектураға шолу
Table 1 – Architecture overview

Кезең	Енгізу	Қабат	Шығару
1-кезең	Бағаналы блок	Тығыз қабат	224×224×3
2-кезең	Тығыз қабат	Өтпелі қабат	28×28×128
3-кезең	Бағаналы блок	Тығыз қабат	14×14×256
4-кезең	Өтпелі қабат	Тығыз қабат	7×7×512

Біз негізгі модель ретінде DenseNet-41 деп аталатын DenseNet-ке ұқсас желіні құрып жатырмыз. Бұл модель мен түпнұсқа DenseNet арасында екі айырмашылық бар. Біріншісі – бірінші конволюциялық қабаттың параметрлері. Бірінші конволюциялық қабатта 64 емес, 24 канал бар, ядро өлшемі 7×7-ден 3×3-ке дейін өзгертілген. Екіншіден, әрбір тығыз

блоктағы қабаттар саны есептік бюджетке сәйкес реттеледі. Осы бөлімдегі барлық модельдерімізді 120 бөлімнен асатын шағын партияның өлшемі 256 PyTorch үйретеді. ILSVRC 2012 ResNet-те қолданылған жаттығулар мен гиперпараметрлердің көпшілігін орындаймыз.

2-кестеде әр түрлі дизайн нұсқаларының өнімділікке әсері көрсетілген. Осы барлық дизайнды біріктіргеннен кейін жүйеміз Stanford

Dogs-қа 79,25% дәлдікке жететінін көреміз, бұл DenseNet-41-ге қарағанда дәлдігі бойынша 4,23% жоғары, есептеу шығындары аз.

Кесте 2 – NVIDIA TX2-тегі жылдамдығы (неғұрлым үлкен болса, соғұрлым жақсы) эталондық құрал NVIDIA TensorRT4.0 кітапханасымен жасалған

Table 1 – The speed (the higher the better) test tool in NVIDIA TX2 is designed with the NVIDIA TensorRT4.0 library

Модель	Top-1 дәлдігі ILSVRC201 224×224	FLOPs 224×224	Жылдамдық (суреттер секундына)
MobileNet V2	70.6	569 M	136.2
ShuffleNet	70.9	524 M	110
Біздің модель	72.1	508 M	240.3

Нақты құрылғылардағы жылдамдық.

FLOP есебі (көбейту-жинақтау операцияларының саны) есептеу құнын өлшеу үшін кеңінен қолданылады. Алайда, нақты уақыттағы уақытқа әсер етуі мүмкін басқа да көптеген факторлар бар екенін ескере отырып, ол нақты құрылғылардағы жылдамдықты тексеруді алмастыра алмайды, мысалы, кәштеу, енгізу-шығару, аппараттық құралдарды оңтайландыру және т.с.с. Бұл бөлім Android-тағы тиімді модельдердің жұмысын бағалайды. Жылдамдық 100 өлшемі бар 100 кескінді өңдеудің орташа уақыты ретінде есептеледі. Біз 100 кескінді 10 рет бөлек жүргіземіз және уақытты орта есеппен аламыз. 2-кестеден көріп отырғаныңыздай, бұл жүйе TX2-дегі MobileNet және MobileNetV2-ге қарағанда әлдеқайда жылдам. MobileNetV2 300 FLOP көмегімен жоғары дәлдікке қол жеткізгенімен, модельдің нақты жылдамдығы 569 FLOP арқылы MobileNet-тен төмен.

Android 9-да жүйеміз кішігірім енгізу өлшемдері үшін MobileNet-тен баяу, бірақ үлкен өлшемдер үшін MobileNet-тен жылдамырақ. Android-тағы жағымсыз нәтиженің екі себебі болуы мүмкін. Бірінші себеп CoreML-ге байланысты, ол Android Metal API-нің жоғарғы жағында салынған. Металл - бұл CNN үшін арналмаған 3D графикалық API. Онда тек 4 арна деректері болуы мүмкін

(бастапқыда RGBA деректерін сақтау үшін қолданылады). Жоғары деңгейлі API құбырды 4-ке кесіп, әр кесудің нәтижесін кәштеуі керек. Бөлінетін конволюция бұл механизмнен тұрақты конволюцияға қарағанда көбірек пайда табуы мүмкін. Екінші себеп - бұл біздің жүйенің архитектурасы. Жүйе көп салалы және тар арналық стильде 113 конволюциялық қабаттан тұрады

Біз жүйенің нақты құрылғылардағы нақты шығу жылдамдығын бағалаймыз. Жылдамдық тестілеу құралымен 100 кескінді өңдеудің орташа уақыты негізінде есептеледі. Бұл уақыт кескіннің алдын-ала өңдеу уақытын қамтиды, бірақ өңдеуден кейінгі бөліктің уақытын қамтымайды (шекті қораптарды декодтау және максималды емес басуды орындау). Әдетте, кейінгі өңдеу процессорында орындалады, ол мобильді GPU-да жұмыс жасайтын басқа бөліктермен асинхронды түрде жұмыс істей алады. Сондықтан нақты жылдамдық біздің сынақ нәтижелеріне өте жақын болуы керек. Pelee-дің қалдық болжау қондырғысы есептеу шығындарын арттырса да, біздің жүйе Android 9-да SSD + MobileNet және FP32 режимінде TX2-ге қарағанда жылдамырақ. 4-кестеден көріп отырғанымыздай, Pelee FP16 режимінде SSD + MobileNet және SSDLite + MobileNetV2-ге қарағанда жылдамдықтың үлкен артықшылығына ие.

Бұл бөлім SSD үшін біздің объектіні анықтау және оңтайландыру жүйесін ұсынады. Біздің оңтайландырудың басты мақсаты - жылдамдықты қолайлы дәлдікпен арттыру. Соңғы бөлімде ұсынылған мүмкіндіктерді алудың тиімді желісін қоспағанда, біз объектілерді анықтау желісін түпнұсқа SSD-ден басқаша жолмен құрдық, бес функция масштабындағы карталармен мұқият өңдедік. Сонымен қатар, анықтау үшін қолданылатын әр мүмкіндік картасы үшін біз болжам жасамас бұрын қалдық блок құрамыз.

Есептеу шығындарын азайту үшін объектілердің санаттары мен шектелетін қораптардың орналасуын болжау үшін біз кіші конволюциялық ядроларды қолданамыз. Сонымен қатар, біз жаттығулардың мүлдем басқа гиперпараметрлерін қолданамыз. Бұл жарналар оқшауланған болып көрінуі мүмкін, бірақ соңғы жүйе PASCAL VOC2007 үшін 70,9% MAP және MS COCO деректер жиынтығы үшін 22,4 MAP жетеді. COCO-дағы нәтиже жоғары дәлдікпен, есептеу шығындарымен 13,6 есе төмен және модельдердің өлшемдерімен 11,3 есе аз YOLOv2-ден асып түседі.

Біздің жүйеде болжам жасау үшін қолданылатын 5 карталар масштабы бар: 19×19, 10×10, 5×5, 3×3 және 1×1. Біз жылдамдық пен дәлдік арттыру үшін 38×38 карта қабатын қолданбаймыз. 19×19 мүмкіндік картасы екі әдепкі терезенің масштабына біріктірілген, ал қалған 4 мүмкіндік карталарының әрқайсысы әдепкі

өрістің бір масштабына біріктірілген. Huang және басқалар [6] сонымен қатар SSD-ді MobileNet-пен біріктіру кезінде 38×38 масштабтағы мүмкіндіктер картасын пайдаланбайды. Алайда, олар болжамға пайдаланылатын 6 карталар масштабын сақтау үшін тағы 2×2 мүмкіндік картасын қосады, бұл біздің шешімімізден өзгеше.

Біздің нысанды анықтау жүйесі SSD3 бастапқы кодына негізделген және Caffe көмегімен оқытылған [20]. Топтаманың мөлшері 32-ге тең. Оқу жылдамдығы бастапқыда 0,005-ке теңестірілді, содан кейін ол сәйкесінше 80к және 100к болды. Қайталаудың жалпы саны 120К құрайды.

Қорытынды: Тереңдіктен бөлінетін конволюция тиімді модель құрудың жалғыз жолы емес. Тереңдіктен бөлінетін конволюцияны қолданудың орнына, біздің ұсынған жүйе кәдімгі конволюцияны қолданып жасалған және ILSVRC 2012, VOC 2007-де сенімді нәтижелерге қол жеткізді. Тиімді архитектуралық дизайнды мобильді графикалық процессормен және аппараттық оңтайландырылған жұмыс уақытының кітапханаларымен үйлестіре отырып, біз мобильді құрылғылардағы суреттерді жіктеу және объектілерді анықтау бойынша нақты уақыттағы болжамды орындай аламыз. Мысалы, біздің ұсынған нысанды анықтау жүйесі Android 9-да 23,6 FPS және NVIDIA TX2-де 125 FPS жоғары дәлдікпен анықтауға мүмкіндік береді.

ӘДЕБИЕТТЕР

- [1] Andrew G Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam. Mobilenets: Efficient convolutional neural networks for mobile vision applications. <https://arxiv.org/abs/1704.04861> (15.10.2020).
- [2] Xiangyu Zhang, Xinyu Zhou, Mengxiao Lin, and Jian Sun. Shufflenet: An extremely efficient convolutional neural network for mobile devices. <https://arxiv.org/abs/1707.01083> (15.10.2020).
- [3] Barret Zoph, Vijay Vasudevan, Jonathon Shlens, and Quoc V Le. Learning transferable architectures for scalable image recognition. <https://arxiv.org/abs/1707.07012> (16.10.2020).
- [4] Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. Mobilenetv2: Inverted residuals and linear bottlenecks // In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. – 2018. – P. 4510-4520.
- [5] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions // In The IEEE Conference on Computer Vision and Pattern Recognition (CVPR). – 2015. – P. 1-9.
- [6] Jonathan Huang, Vivek Rathod, Chen Sun, Menglong Zhu, Anoop Korattikara, Alireza Fathi, Ian Fischer, Zbigniew Wojna, Yang Song, Sergio Guadarrama, et al. Speed/accuracy trade-offs for modern convolutional object detectors. <https://www.arxiv-vanity.com/papers/1611.10012/> (17.10.2020).

- [7] Gao Huang, Zhuang Liu, Kilian Q Weinberger, and Laurens van der Maaten. Densely connected convolutional networks. <https://arxiv.org/pdf/1608.06993.pdf>? (21.10.2020).
- [8] Aditya Khosla, Nityananda Jayadevaprakash, Bangpeng Yao, and Fei-Fei Li. Novel dataset for fine-grained image categorization: Stanford dogs // In Proc. CVPR Workshop on Fine-Grained Visual Categorization (FGVC). – 2011. – Vol. 2. – P. 1.
- [9] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database // In Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference. – P. 248-255.
- [10] Christian Szegedy, Sergey Ioffe, Vincent Vanhoucke, and Alexander A Alemi. Inception-v4, inception-resnet and the impact of residual connections on learning // In AAAI. – 2017. – P. 4278-4284.
- [11] Zhiqiang Shen, Zhuang Liu, Jianguo Li, Yu-Gang Jiang, Yurong Chen, and Xiangyang Xue. Dsod: Learning deeply supervised object detectors from scratch // In The IEEE International Conference on Computer Vision (ICCV). – 2017. – Vol. 3. – P. 7.
- [12] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift // In International Conference on Machine Learning. – 2015. – P. 448-456.
- [13] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C Berg. Ssd: Single shot multibox detector // In European conference on computer vision. – 2016. – P. 21-37.
- [14] Mark Everingham, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. The pascal visual object classes (voc) challenge // International journal of computer vision. – 2010. – P. 303-338.
- [15] Joseph Redmon and Ali Farhadi. Yolo9000: better, faster, stronger. <https://arxiv.org/pdf/1612.08242.pdf> (26.10.2020).

REFERENCES

- [1] Endryu Dzhi Khovard, Menglong Chzhu, Bo Chen, Dmitriy Kalenichenko, Veydzhun Van, Tobias Veyand, Marko Andreyetto i Khartvig Adam. *Mobil'nyye seti: effektivnyye svertochnyye neyronnyye seti dlya prilozheniy mobil'nogo zreniya* [in USA: Mobilenets: Efficient convolutional neural networks for mobile vision applications]. <https://arxiv.org/abs/1704.04861> (15.10.2020).
- [2] Syanyuy Chzhan, Sin'yu Chzhou, Mensyao Lin' i TSzyan' Sun'. *Shufflenet: chrezvychayno effektivnaya svertochnaya neyronnaya set' dlya mobil'nykh ustroystv* [in China: Shufflenet: An extremely efficient convolutional neural network for mobile devices]. <https://arxiv.org/abs/1707.01083> (15.10.2020).
- [3] Barret Zof, Vidzhay Vasudevan, Dzhonaton Shlens i Kuok V. Le. *Izucheniye perenosimyykh arkhitektury dlya masshtabiruyemogo raspoznavaniya izobrazheniy* [in USA: Learning transferable architectures for scalable image recognition]. <https://arxiv.org/abs/1707.07012> (16.10.2020).
- [4] Mark Sandler, Endryu Khovard, Menglong Chzhu, Andrey Zhmoginov i Lyan-Chi Chen. *Mobilenetv2: obratnyye ostatki i lineynyye uzkiye mesta* [in Great Britain: Mobilenetv2: Inverted residuals and linear bottlenecks]. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. – 2018. – P. 4510-4520.
- [5] Kristian Segedi, Vey Lyu, Yantsin TSzya, P'yer Sermane, Skott Rid, Dragomir Angelov, Dumitru Erchan, Vinsent Vankhauk i Endryu Rabinovich. *Uglublyayas' v izviliny* [in USA: Going deeper with convolutions]. In The IEEE Conference on Computer Vision and Pattern Recognition (CVPR). – 2015. – P. 1-9.
- [6] Dzhonatan Khuang, Vivek Ratod, Chen Sun, Menglong Chzhu, Anup Korattikara, Alireza Fatkhi, Yan Fisher, Zbignevo Voyna, Yan Song, Serkhio Guadarrama i dr. *Kompromiss mezhdu skorost'yu i tochnost'yu dlya sovremennykh detektorov svertochnyykh ob'yektov* [in China: Speed/accuracy trade-offs for modern convolutional object detectors]. <https://www.arxiv-vanity.com/papers/1611.10012/> (17.10.2020).
- [7] Gao Khuan, Chzhuan Lyu, Kilian K. Vaynberger i Lorens van der Maaten. *Plotno svyazannyye svertochnyye seti* [in USA: Going deeper with convolutions]. <https://arxiv.org/pdf/1608.06993.pdf>? (21.10.2020).
- [8] Adit'ya Kkhosla, Nit'yananda Dzhayadevaprakash, Bangpeng Yao i Fey-Fey Li. *Novyy nabor dannykh dlya detal'noy kategorizatsii izobrazheniy: sobaki Stenfordskogo universiteta* [in Great Britain: Novel dataset for fine-grained image categorization: Stanford dogs]. In Proc. CVPR Workshop on Fine-Grained Visual Categorization (FGVC). – 2011. – Vol. 2. – P. 1.
- [9] TSzya Den, Vey Dong, Richard Socher, Li-TSzya Li, Kay Li i Li Fey-Fey. *Imagenet: krupnomasshtabnaya baza dannykh iyerarkhicheskikh izobrazheniy* [in USA: Imagenet: A large-scale hierarchical image database]. In Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference. – P. 248-255.
- [10] Kristian Segedi, Sergey Ioffe, Vinsent Vankhauk i Aleksandr Alemi. *Inception-v4, inception-resnet i vliyaniye ostatochnyykh soyedineniy na obucheniye* [in USA: Inception-v4, inception-resnet and the impact of residual connections on learning]. In AAAI. – 2017. – P. 4278-4284.

- [11] Chzhitsyan Shen, Chzhuan Lyu, TSzyan'go Li, YU-Gan TSzyan, Yuzhun Chen i Syan"yan Syue. *Dsod: Izucheniye detektorov ob"yektov s glubokim kontrolem s nulya* [in China: Dsod: Learning deeply supervised object detectors from scratch]. In The IEEE International Conference on Computer Vision (ICCV). – 2017. – Vol. 3. – P. 7.
- [12] Sergey Ioffe i Kristian Segedi. *Paketnaya normalizatsiya: uskoreniye glubokogo obucheniya seti za schet umen'sheniya vnutrennego kovariantnogo sdviga* [in USA: Batch normalization: Accelerating deep network training by reducing internal covariate shift]. In International Conference on Machine Learning. – 2015. – P. 448-456.
- [13] Vey Lyu, Dragomir Anguelov, Dumitru Erhan, Kristian Segedi, Skott Rid, Cheng-Yan Fu i Aleksandr S. Berg. *Ssd: odnorazovyy mul'tiboks-detektor* [in USA: Ssd: Single shot multibox detector]. In European conference on computer vision. – 2016. – P. 21-37.
- [14] Mark Everingem, Lyuk Van Gul, Kristofer K.I. Uil'yams, Dzhon Vinn i Endryu Zisserman. *Zadacha klassov vizual'nykh ob"yektov paskal' (vokal)* [in Great Britain: The pascal visual object classes (voc) challenge]. International journal of computer vision. – 2010. – P. 303-338.
- [15] Dzhozef Redmon i Ali Farkhadi. *Yolo9000: luchshe, bystreye, sil'neye.* [in USA: Yolo9000: better, faster, stronger]. <https://arxiv.org/pdf/1612.08242.pdf> (26.10.2020).

ТЕРЕҢ ОҚЫТУ НЕГІЗІНДЕ НАҚТЫ УАҚЫТТА ОБЪЕКТІЛЕРДІ ТАҢУ

Қуандықов Абу Абдикадырович, т.ғ.д., Қожа Ахмет Яссауи атындағы Халықаралық қазақ-түрік университеті, Түркістан, Қазақстан, abu.kuandykov@mail.ru

Тасполат Еркебулан Айдарбекулы, магистрант, Қожа Ахмет Яссауи атындағы Халықаралық қазақ-түрік университеті, Түркістан, Қазақстан, yerkebulantaspolat@gmail.com

РАСПОЗНАВАНИЕ ОБЪЕКТОВ В РЕАЛЬНОМ ВРЕМЕНИ НА ОСНОВЕ ГЛУБОКОГО ОБУЧЕНИЯ

Қуандықов Абу Абдикадырович, д.т.н., Международный Казахско-турецкий университет имени Ходжи Ахмеда Яссауи, г.Туркестан, Казахстан, abu.kuandykov@mail.ru

Тасполат Еркебулан Айдарбекулы, магистрант, Международный Казахско-турецкий университет имени Ходжи Ахмеда Яссауи, г.Туркестан, Казахстан, yerkebulantaspolat@gmail.com

Аннотация. В статье разрабатывается ряд эффективных архитектур, которые решают вопросы увеличения исследований по эффективному моделированию мобильных устройств с ограниченными вычислительными мощностями и ресурсами памяти, что увеличивает потребность в использовании сверточной нейронной сети (CNN). Среди них архитектуры MobileNet, ShuffleNet и MobileNetV2, предложенные в последние годы. В исследовании делается вывод, что все эти модели основаны на глубокой изоляции, которая неэффективно реализована во многих системах глубокого обучения. Поэтому в данной работе мы предложили эффективную систему постоянной свертки. Затем мы предлагаем систему обнаружения объектов в реальном времени путем интеграции системы с детектором Single Shot MultiBox (SSD).

Ключевые слова: сверточная нейронная сеть, глубокое обучение, обнаружение объектов, эффективное моделирование, абляция.

The Bulletin of Kazakh Academy of Transport and Communications named after M. Tynyshpayev
ISSN 1609-1817. Vol. 116, No.1 (2021), pp.293-299

UDC 005.9

10.52167/1609-1817-2020-116-1-293-299

ANALYSIS OF THE MANAGEMENT OF MATERIAL AND TECHNICAL ASSETS OF ENTERPRISES

Aizhan Sirkebay, Master student, International Information Technology University, Almaty, Kazakhstan; asirkebay@gmail.com

Abstract. Variability of demand for goods, depending on their availability and cost is a frequently observed phenomenon in the sphere of production and consumption. This paper discusses the possibility of using the ABC analysis for the inventory management system. This