Lugemisülesanne
(EBSCO Discovery; refereerimine ja viitamine)

Esita fail või link Moodle'is hiljemalt 17. novembril 2024.

NB! Väljaspool TÜ arvutivõrku pääsete andmebaasile ligi siit: https://utlib.ut.ee/andmebaasid?title=E > valige EBSCO Discovery nime kõrval oleval sinisel linnukesel > valige "Otselink andmebaasi" ning sisenege TÜ kasutajatunnustega.

Nimi: Joonas Tiitson Eriala: Informaatika Bsc

1. **Tehke otsingud EBSCO Discoverys**

Otsingu teema:

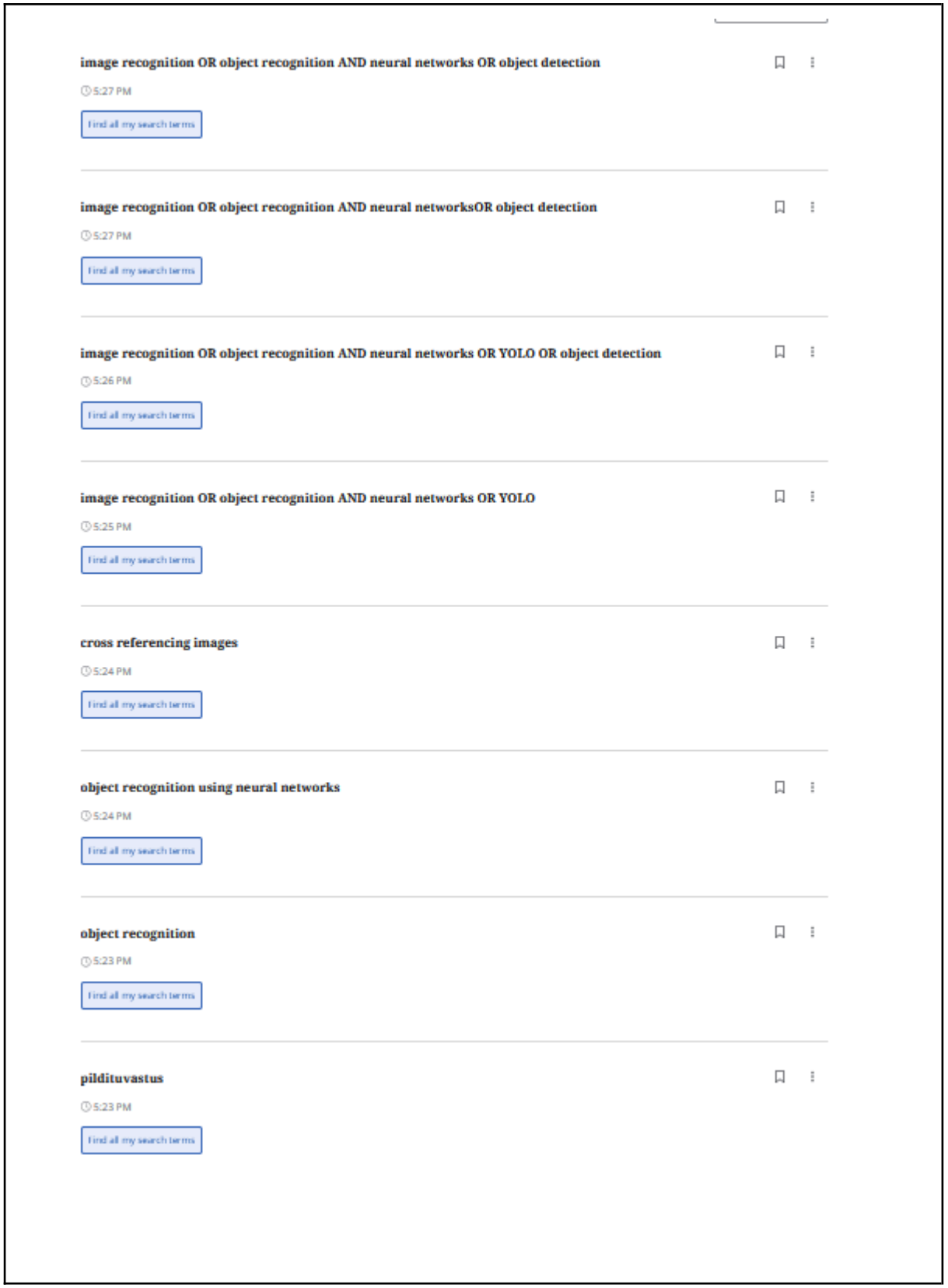
Koostage õppematerjali eeskujul päring või mitu päringut oma teema (lõputöö või vikiartikli) kohta. Kasutage vastavalt vajadusele loogikaoperaatoreid, jutumärke, kärpimist ning sulge. Kaasake sünonüüme ja teisi teemaga seotud sõnu.

Päring(ud) (s.o laiemad ja kitsamad päringud või ainult üks päring, mille järgi hakkate oma teema jaoks kirjandust otsima):

object recognition object recognition using neural networks image recognition OR object recognition AND neural networks OR YOLO OR object detection image recognition OR object recognition AND neural networks OR object detection

1.1. Valige "Advanced Search" ning sisestage päring. Operaatoritega OR ühendatud sünonüümid sisestage ühele reale ja read ühendage operaatoriga AND.
1.2. Katsetage päringu muutmist ja kasutage vastavalt otstarbekusele erinevaid piiranguid, (märksõnad, ilmumisaeg, erinevad otsinguväljad jm). Märksõnu, mille järgi otsida, leiate sõnaotsinguga leitud relevantsetest kirjetest.
Õppematerjal "Otsing EBSCO Discoverys".

Tehke vähemalt kaks erinevat otsingut. Klõpsake otsingukasti all nuppu "Search History" (kui see pole juba avatud), märkige ära 2 parimat otsingut, klõpsake "Print Search History", kopeerige ja kleepige tulemused oma faili. Võib ka esitada kuvatõmmise.



2. **Otsingute võrdlus**

Võrrelge oma otsinguid ja analüüsige nende tulemuslikkust. Kuidas erinevad piirangud tulemusi mõjutasid? Kas leidsite teemakohaseid artikleid? Esitage selgitused ja analüüs.

Lihtsamatel otsingutel tekib probleem et tulemused on väga laiad, kuigi pildituvastus ei ole just väga mitmeti mõistetav ala tekib ikka küsimus et mis on minul tarvis. Milline peaks olema minu mudel kui ma tahan PCB-de pealt võimalikult palju tarvilikku infot saada. Viimane otsing andis mulle hea tulemuse, kuna üks probleemidest mis tahan lahendada on pildilt komponentide orientatsiooni tuvastamine, ning leidsin artikli pealkirjaga “Oriented object detection in satellite images using convolutional neural network based on ResNeXt.”, mis arvan et tuleb vägagi kasuks oma probleemi lahendades. Aga ka üldiselt vaadates saab häid ideid ja ettekujutuse kuidas oma tulevikus enda projektiga tegeleda. Ajavahemikke märkides on ka kerge leida kaasaegseid artikleid, et näha mis kõige uuem ja parem lahendus on.

3. Refereerimine

1. Kirjutage allolevasse tühja kasti u 150-200-sõnaline refereering leitud allikate (vähemalt 2) põhjal. Oluline on allikates olevat infot sünteesida oma teksti. Refereering võiks vastata küsimusele – mille kohta saab leitud allikatest ülevaate?

2. Refereering

- koosneks ühest lõigust;

- koosneks 150–200 sõnast;

- oleks sidus, st tekst peaks moodustama sisulise ja keelelise terviku;

- vastaks refereerimisreeglitele, sh sisaldaks tekstisiseseid viiteid ja lõpus viidete allikakirjeid.

3. Lisa ekraanitõmmise või foto kujul **ainult** see **täpne** tekstiosa, mida oled refereerinud.

Oma uurimistöös plaanin kasutada YOLO mudelit et tuvastada PCB pealt erinevaid komponente. Kuid mõnel komponendil on oluline selle suund. Kuna YOLO (you only look once) mudelid ei saa mitu korda pilti läbida siis tekib probleem selle ülesandega, ja tuleb kasuks kaheastmeline mudel. Algseks objekti tuvastamiseks võib ikka kasutada YOLO mudelit, aga suuna määramiseks saab lisada teise astme. Kuid selline kaheastmeline tegevus on arvutustelt keeruline ja on aeglasem kui üheastmeline meetod, kasutades muu arhitektuuri mudelit, näiteks ResNeXt101. Objekti ja selle suuna tuvastamiseks on vaja kõigepealt märkida selle asukoht horisontaalse piirdekasti (*HBB*) ja suund orienteeritud piirdekastiga (*OBB*). (Haryono 2023) YOLO mudel on aga väga kasulik kui pildilt on vaja lihtsalt tuvastada kiirelt otsitavaid komponente, sellepärast jääb see igatahes minu töös esimeseks sammuks. YOLO mudel on nii kiire kuna kasutab väheseid kõrvalisi teeke ning on kirjutatud otse C keeles ja CUDA-s. (Yang 2023) Kuigi ResNeXt101 saavutab häid tulemusi kiire tuvastusega ja on kiirem kaheastmelisest mudelist sellisel juhul, siis on mul

tarvis vaheldumisi kasutada suuna tuvastust. Mõnel komponendil pole nii täpne tuvastus lihtsalt vajalik. YOLO on praegu ideaalne kesktee kiirusest ja täpsest tuvastamisest, et oleks kasulik tööstuses.

1) Haryono, A.; Jati, G.; Jatmiko, W. Oriented object detection in satellite images using convolutional neural network based on ResNeXt. ETRI J. 2023.
2) Yang, C-H; Lin, Y-N; Wang S-K; Shen, V.R.L; Tung, Y-C; Lin J-F. An Edge Computing System for Fast Image Recognition Based on Convolutional Neural Network and Petri Net Model. Springer Nature. 2023

2.2 Real-time object detection

The YOLO series are neural-network-based algorithms for object detection. YOLO uses a CNN architecture with 24 convolutional layers and 2 fully connected layers. Its total computational cost is lower than a fully connected neural network. Also, YOLO is less dependent on peripheral libraries. Written in *C* language and *CUDA*, it can be used without installing many dynamic-link libraries [25].

A state-of-the-art method that uses an oriented object detector based on a CNN is the box-boundary-aware vector (BBAVector) [16], which is adapted from the U-Net architecture [17]. The BBAVector method performs object detection by expanding a horizontal-keypoint-based object detector into oriented object detection. BBAVectors are distributed on the four quadrants of a Cartesian coordinate system to solve angular problems. The BBAVector method performs feature extraction using a ResNet101 encoder to capture semantic information in the shallow layer. However, ResNet only has a sequential layer on its building blocks, being less effective at extracting features during downsampling.

To improve the performance of the BBAVector method for oriented object detection, this study has the following contributions:

1. The method is modified by replacing the encoder with a block from ResNeXt101. With its cardinality given by a stack of parallel layers with a homogeneous design and multibranch architecture with few hyperparameters, ResNeXt101 is more effective at capturing information than ResNet101, which is originally used in the BBAVector method.
2. The input image size is modified from 608 × 608 pixels to 416 × 416 pixels. This size change affects the performance results, with experimental results demonstrating improved performance.

The remainder of this paper is organized as follows. Section 2 describes related work on object detectors, bounding boxes, ResNet versus ResNeXt, and oriented object detection. Section 3 details the proposed architecture and research methods. Section 4 presents the experimental setup for detailed testing and evaluation metrics. Section 5 reports and discusses the results. Finally, Section 6 presents the conclusions.

2 | RELATED WORK

2.1 | Object detectors

Object detectors are generally divided into single- and two-stage methods [18].

A one-stage detector classifies and localizes semantic objects in a single shot using dense sampling. This method uses predefined boxes (keypoints) of various scales and ratios to localize and enclose objects. A single-stage detector is YOLO, which improves feature extraction by adding convolutional layers at different locations. In addition, a residual module is added to overcome the gradient loss related to the network depth during training. YOLO enables the detection of small objects in real time.

A two-stage detector has a separate module to generate a region of interest (RoI) (proposal interest). A two-stage method is Fast RCN, which attempts to find several variable proposal objects in an image in the first stage and then classifies and localizes the objects in the second stage. Given the two separate stages, proposal generation is usually time-consuming because of the complex architecture and lack of global context. Therefore, a one-stage detector outperforms a two-stage detector regarding the detection speed and architecture simplicity.

2.2 | Bounding boxes

Two types of bounding boxes are used in object detection: HBB and OBB [19]. The HBB is represented by points at the top-left and bottom-right positions to form a horizontal box. An OBB, also called rotating bounding box, is described by the center coordinates, width, height, and orientation. An HBB implementation is intended to detect vehicles in aerial imagery using a feature-balanced pyramid network (FBPN) in [20], which achieves high-performance feature extraction. The FBPN is combined with faster RCNN, which is modified by adopting a focal loss function to reduce the imbalance between complex and easy samples to promote detection of small objects. Other HBB implementations include the detection of vehicle number plates [21] and human–object interactions [22].

Applications of OBBs include the semi-anchor-free detector (SAFDet), which uses a rotation-anchor-free branch to increase foreground features through regression [23]. Using a RoI transformer and attention model, SAFDet can overcome the problems of misaligned horizontal proposals and complex background interference. A center prediction module is also introduced to increase object localization and suppress background noise. The combined rotation-anchor-free branch and center prediction module reduce the computational cost. Experimental results show that SAFDet can achieve high accuracy and fast detection.

2.3 | ResNet versus ResNeXt

An essential factor in the performance and computational speed of a method is the architecture of the underlying model. Several types of backbone architectures have been implemented in deep learning, including ResNet and ResNeXt, for object detection [18]. A deeper CNN is more complex and difficult to train because