

Tartu Ülikool  
Arvutiteaduse instituut  
Informaatika õppekava

**Joonas Tiitson**

**PCB komponentide tuvastamine**

**Bakalaureusetöö**

Juhendaja: Kristo Raun, PhD

Tartu 2025

# Sisukord

|  |    |
|--|----|
| 1. Sissejuhatus.....   | 3  |
| 2. Mõisted, terminid ja lühendid.....  | 4  |
| 3. Kirjanduse ülevaade.....  | 5  |
| 3.1 Pildituvastus mudelite ülevaade.....   | 5  |
| 3.2 YOLO võrreldes teiste mudelitega.....  | 6  |
| 3.3 Pool juhendatud õpe.....   | 7  |
| 3.4 <i>Zero-Shot</i> õpe.....  | 8  |
| 3.5 Suuna tuvastus Yolo mudeliga.....  | 8  |
| 4. Metoodika.....  | 10 |
| 4.1 Andmekogumi loomine.....   | 10 |
| 4.2 Andmehulga märgendamine.....   | 10 |
| 4.3 YOLO tulemused.....  | 11 |
| 4.4 FasterRCNN tulemused.....  | 11 |
| 4.5 Ülevaade praktilisest rakendusest.....   | 11 |
| 5. Tulemuste analüüs.....  | 12 |
| 5.1 Kaamera põhine kontroll.....   | 12 |
| 5.2 ResNeXt võrreldes YOLO mudeliga.....   | 12 |
| 5.3 Mudelite nõudlikkus.....   | 12 |
| 7. Tulemuste arutelu.....  | 13 |
| Järgnevad peatükid arutlevad, mida saaks tulevikus edasi arendada. Samuti toovad need välja mõningad probleemid, mille lahendamiseks oleks vaja erinevalt tulevikus läheneda, et edasised loodud lahendused oleksid täpsemad ja kasulikumad..... | 13 |
| 7.1 Märgendatud andmekogu.....   | 13 |
| Viidatud kirjandus.....  | 14 |

## 1. Sissejuhatus

Elektroonika tööstuses toodetakse sageli trükkplaate. Trükkplaatide peal on erinevaid elektroonika komponente, nende hulgas näiteks transistorid, takistid. Komponendid on üsna väikesed ja neid on iga trükkplaadi peal palju, sellepärast on nende korrasoleku hindamine täpne ning rutiinne töö. Siit tulenevalt saakski appi võtta modernsed masinõppel baseeruvad lahendused. Kuna arvutid ei väsi ära vaadates pilte. Eesmärgiks ongi luua mudel mis on täpne ning usaldusväärne trükkplaadi komponentide tuvastamisel. Andes ette trükkplaadil leiduvate komponentide kogumi ja trükkplaadi pildi peaks lahendus ütlema mis on puudu. Mõne komponendi puhul on oluline ka tuvastada selle orientatsiooni.

Tähtis on ka trükkplaatide kiire analüüsimine, müra eiramine ning väikeste komponentide märkimine, mis teeb lahenduste loomise keeruliseks. Ei saa kasutada suure analüüsimis ajaga meetodeid, mis aeglustaks kogu tootmise järke. Ega ei saa ka kasutada väga kiireid kuid tihti vigu tegevaid mudeleid ega meetodeid. Peab leidma keskte, nii et vähemalt 90% vigastest plaatidest märgitaks ära ning 90% tervetest plaatidest pääseks läbi ilma et neid valesti katkiseks märgitaks.

Kuna ideaalis peaks olema loodud tarkvara vabalt ja tasuta kasutatav kõigile, kes seda tahavad siis võiksid olla ka kõik lisa rakendused ja masinõppe mudelid tasuta kasutatavad kõigile.

## 2. Mõisted, terminid ja lühendid

*Zero-Shot learning* – Masinõppes kasutatav meetod, kus mudel kasutab enne olemas olevaid seoseid, et klassifitseerida uut infot.

*Printed circuit board* – Trükkplaat, elektroonikas kasutatav montaažiplaat, millele on võimalik paigaldada elektroonikakomponendid ja need elektriliselt ühendada.

*Neural network* – Eesti keeles närvivõrk, arvuti süsteem, mis on modelleeritud inim aju ja bioloogilise närvivõrgu järgi

*Support vector machine* – vektormasin, mis on loodud andmete analüüsimiseks ning klassifitseerimiseks, kasutades suurima erinevuse leidmist

*Bounding box* – piirkast, kasutatakse mingi ala märkimiseks.

*UI – user interface*, eesti keeles kasutajaliides, visuaalne element rakendusel et võimaldada seda kasutada.

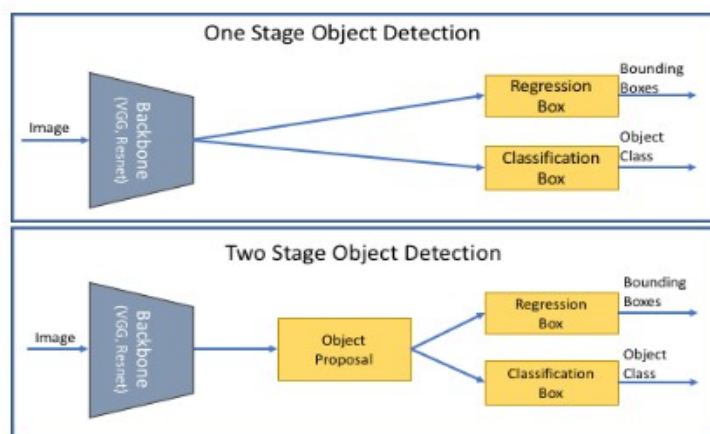
*Convolutional neural network – CNN*

### 3. Kirjanduse ülevaade

Järgnevates alampeatükis on toodud välja seletus mitmetele masinõppel põhinevate objektituvastuste metoodikate ülesehitusele. Samuti käsitlen ka andmehulga loomiseks kasulikke võtteid, mis tulevikus saavad vähendada manuaalset tööd edasisel arendusel.

#### 3.1 Pildituvastus mudelite ülevaade

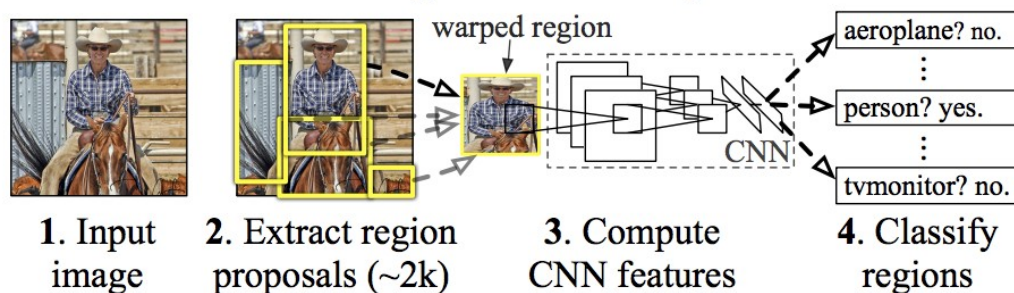
Laialdaselt on kasutuses tänapäeval kahe erineva struktuuriga pildituvastus mudelit (Andreu 2018). Nendeks on ühe astmelised objekti tuvastus mudelid nagu näiteks YOLO (*you only look once*) ning kahe astmelised objekti tuvastus mudelid nagu näiteks R-CNN (*Region-based Convolutional Neural Network*) (Andreu 2018). Nende peamine erinevus tuleneb sellest, et kahe astmelised mudelid kõigepealt valivad huvialad millelt tuvastada ja seejärel klassifitseerivad leitud ala, kuid üheastmelised mudelid teevad mõlemat sammu korraga (Vaata joonis 1).



(Joonis 1. ühe- ja kaheastmelise mudeli üldine struktuur, Andreu 2018)

Järgnev seletus R-CNN tööpõhimõttest, toetub peamiselt „14.8. Region-based CNNs (R-CNNs)” peatükile, raamatust „Dive into deep learning” (Zhang 2024). R-CNN mudelid töötavad mitmeastmeliselt, tuvastuseks läbivad terve pildi, kust valivad mitu huvi pakkuvat regiooni. Neid regioone omakorda muudetakse, nii et oleksid vajalike mõõtmetega närvivõrgus (*neural network*) töötlemiseks. Igat tüüpi objekti jaoks luuakse toetav vektormasin (*Support vector machine*), mis määrab kas valitud alas on selle vektormasina tuvastatav objekt. (Vaata joonis 2)

## R-CNN: *Regions with CNN features*

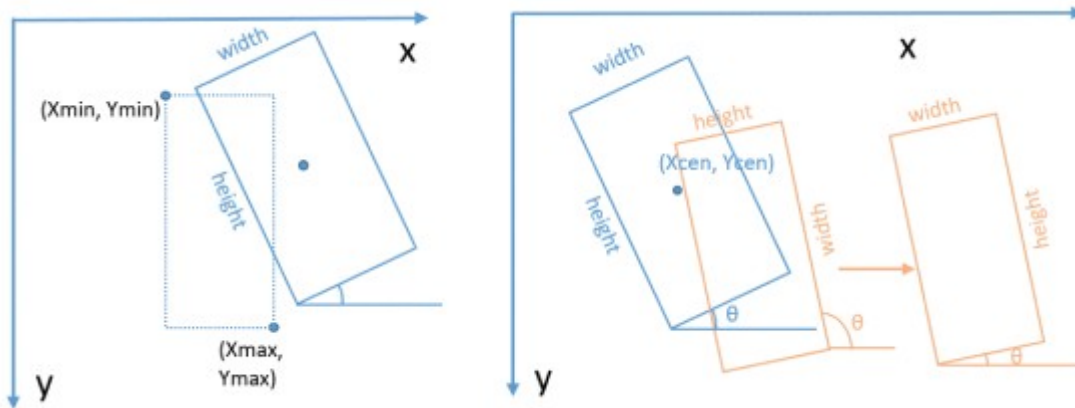


(Joonis 2. RCNN mudeli ülesehitus, Girshick 2014)

Kui kõik vektormasinad on oma tulemused saanud tuvastatava ala kohta, antakse need edasi lineaar regressiooni mudelile, mille abil on võimalik lõpuks leida mis objekt asub vaatluse all oleval alal. Nii tehakse iga algselt huvi pakkunud alaga. Kuna töötlemine on mitmeastmeline, siis tihti on see väga aeglane ja ressursikulukas. YOLO mudel on uuem ning loodud kiireks pildituvastuseks. Selle struktuur on lihtsam kui R-CNN mudelid, omades ainult ühte närvivõrku, mis töötleb pilti otse (Redmon 2016). Mudel tuvastab samaaegselt piltidelt piirdekaste (*Bounding box*) ning ka ennustab mis objektiga on tegu (Redmon 2016). Seega on mudeli sisendiks pilt millelt on vaja tuvastada objekte ja tulemuseks on pildilt tuvastatud objektide piirdekastid ning klassid. Mudeli eelisteks on selle kiirus kuid võrreldes R-CNN mudelitele on see vähem täpne.

### 3.2 YOLO võrreldes teiste mudelitega

Uurimistöös on kasutusel YOLO mudel et tuvastada PCB pealt erinevaid komponente. Mõnel komponendil on oluline selle suund. Kuna YOLO mudelid ei saa mitu korda pilti läbida siis tekib probleem selle ülesandega, ja tuleb kasuks kaheastmeline mudel (Haryono 2023). Algseks objekti tuvastamiseks võib ikka kasutada YOLO mudelit, aga suuna määramiseks saab lisada teise astme (Haryono 2023). Kuid selline kaheastmeline tegevus on arvutustelt keeruline ja on aeglasem kui üheastmeline meetod, kasutades muu arhitektuuri mudelit, näiteks ResNeXt101 (Haryono 2023). Objekti ja selle suuna tuvastamiseks on vaja kõigepealt märkida selle asukoht horisontaalse piirdekasti (*HBB*) ja suund orienteeritud piirdekastiga (*OBB*) (Haryono 2023).



(Joonis 3. Näide horisontaalsest ja suunatud piirdekastist. Wang 2019)

YOLO mudel on aga väga kasulik kui pildilt on vaja lihtsalt tuvastada kiirelt otsitavaid komponente (Yang 2023). YOLO mudel on nii kiire kuna kasutab väheseid kõrvalisi teeke ning on kirjutatud otse masinkeeledel lähedates C ja CUDA keeles (Yang 2023). ResNeXt ja YOLO lähenevad samale ülesandele eri nurkade alt. Nende struktuuriline ülesehitus on vägagi erinev millega kaasnevad ka eelised kui ka omad puudused. Näiteks peaks YOLO olema küll kiirem kui ResNeXt, kuid see eest ka vähem täpne ning suunatuvastusel halvem.

YOLO mudeli eelisteks veel on andmete säilitamine treenimis protsessi vältel. See aitab ära hoida valede seoste tekkimist mudelis. Samuti aitab andmete alles hoidmine tulevikus mudeli ümbertreenimisel, kui näiteks tahta lisada mingit objekti mida see tuvastaks hiljem juurde (Wang 2024). Tänu GELAN ja PGI objekti tuvastuse koos kasutamisele saavutab YOLOv9 (üheksas versioon YOLO mudelist mis on ka vabavara MIT litsentsiga) kõrgema täpsuse teistest reaalsajas tuvastavatele algoritmidele (Wang 2024).

### 3.3 Pool juhendatud õpe

Pool juhendatud õpet on laiadsemalt kasutatud R-CNN mudelites, kuid YOLO mudelites pole see väga levinud (Zhou 2022). Edu on saavutatud YOLOv5-le õpetaja-õpilase mudeli loomisega. Õpetaja-õpilase mudeli põhimõtteks on kasutada inimmärgendatud andmetel treenitud mudelit

uute treening andmete märgendamisel, pärast ühendada uus märgendatud nii õelda õpetaja treeninghulk ja inimmärgendatud treeninghulk, et treenida õpilas-mudel.

YOLOv5 õpetaja-õpilas mudelis kasutati üle 1300 videot 21-st koolist ühes rajoonis, mis märgendati käsitsi (Zhou 2022). Pärast treeniti mudel nende piltide peal ning kasutati seda mudelit et ära märgistada ka teisest rajoonist pärit videod (Zhou 2022). Tekkinud andmekogumi kasutati, et omakorda treenida veel üks uus YOLOv5 mudel, mis saavutas testhulgal parema tulemuse kui algne mudel (Zhou 2022). Pool juhendatud õpe on kasulik et luua täpsemaid mudeleid ning vähendada andmekogumi loomiseks kulutatavat aega.

### **3.4 Zero-Shot õpe**

*Zero-Shot* õpe põhimõte võtab eeskujuks inimlaadse õppimise (IEEE 2008). Põhimõte sai ametlikult avaldatud aastal 2008, kuid praktilisi lahendusi ning teste tehti juba varasemalt (IEEE 2008). *Zero-Shot* õppe peamine eesmärk on võimaldada info tuvastamist ja klassifitseerimist, mida eelnevas treening hulgas ei leidunud. Selline lähenemine aitab vähendada manuaalse märgendamise hulka (IEEE 2008). Tuvastatavaid objekte on võimalik läbi üldisemate seletuste või abstraktsemate kujunditega lihtsustada, ning selliste lihtsustuste abil on võimalik ka täiesti uut infot tuvastada. Artiklis *Zero-data Learning of New Tasks* on mainitud näiteks uute kirjamärkide tuvastamist, kui on antud abstraktsem tähe üldine struktuur ja treening hulgas on sarnaselt juba treenitud välja mudel teisi kirjavahe märke seondama sarnastele abstraktsetele kujutistele tekib mudelil võimalus ka uute enne nägemata objekte õigesti tuvastada.

### **3.5 Suuna tuvastus Yolo mudeliga**

Yolo tuvastus üldiselt kasutab eelnevalt mainitud piirdkaste (bounding box), kuid objektide suuna tuvastuseks on kasu võtmepunktidest (*KeyPoints*). Nii saab tuvastada objekti servi ning olenevalt nende asetsusest saab määrata ka objektide suundi. Nii näiteks on võimalik teha paremal ja vasakul asetsevatel munajuhadel ning tuvastada osalise peitumisega objekte (Boonkong 2024). Suurim eelis oli osaliselt peituvate objektide tuvastamine, mis ei peaks selle



projekti puhul probleemi tekitama, kuid nende lahendus kasutada võtme punktidel baseeruvat tuvastust arvatavasti lihtsustaks ka objektide suuna tuvastust.

## 4. Metoodika

Selle peatüki eesmärk on anda lugejale ülevaade loomisprotsessist ning lõplikult valminud rakendusest. Rakenduse eesmärk on lahendada sissejuhatuses püstitatud probleem. Riistvaraliselt on tulemuste loomiseks kasutatud kaamerat mille kaadrisagedus on 30 kaadrit sekundis ning arvutit millel on protsessoriks i5-11400f ja graafikakaardiks RTX 3060.

Kuigi kasutatavad meetodid on väga arvutuslikult nõudlikud, peab süsteem olema vähemalt ümber liigutatav, või siis kaugelt ligipääsetav. Kuna tooteid võidakse filmida mitmes erinevas kohas hoones, siis on mõistlik, kui kogu süsteem on lihtsalt ülesseatav, või video saadetakse ühte kesksesse arvutisse, et seal saaks kaugelt tuvastust kasutada. Miinusteks oleks muidugi suurenenud latentsus, kuid arvutuskiirus mis nii oleks võimalik saavutada korvaks selle ära.

### 4.1 Andmekogumi loomine

Andmekogumiks kasutatakse kombinatsiooni juba eksisteerivatest märgendatud PCB kogumeid, üksik komponentide pilte ja olemasolevate trükkplaatide pilte. Märgenduse teostas töö autor käesoleva töö raames. Andmekogum ise koosneb piltidest ja nendelt vastavalt leitud komponentide loeteluga failidest, mille saab eraldada treening kogumiks ja test kogumiks.

### 4.2 Andmehulga märgendamine

Andmehulk loodi kasutades Pythoni skripti, mis filmis käivitamisel ees olevat trükkplaati, iga kaader filmimisest salvestati eraldi pildina. Sellega pildistati erinevate suundade alt võimalikult palju erinevaid kättesaadavaid trükkplaate. Andmehulga märgendamiseks kasutati CVAT-i. Peamiseks eeliseks CVAT-il oli selle lai märgenduse formaatide valik ning kergesti mõistetav UI. Lõplikuks andmehulgaks valmis 868 pilti, mille peal on märgitud skeleti (*ik Skeleton*) põhiselt kondensaatorite negatiivsed ja positiivsed küljed. Seda andmehulka kasutati ka COCO stiilis OBB sünteesimiseks, mida oli vaja et ResNeXt põhist mudelit treenida.

*OB*B sünteesimiseks kasutati iga skeleti kahte otspunkti, võeti nende kaugus ruudu laiuks ja kõrguseks. Otspunktide vahelise joone keskpunkt võeti ruudu keskpunktiks. Nurk *OB*B kalde saamiseks loodi arktangensi nende otspunktide vastavate koordinaatide vahede alusel. Ehk kaks otspunkti moodustasid täisnurkse kolmnurga hüpotenuusi otspunktid ning nende  $x$  koordinaatide vahe moodustas ühe kaateti ja  $y$  koordinaatide vahe moodustas teise. Nende jagatis on *OB*B nurga tangens, mis muudeti arktangensi abil täielikuks nurgaks.

#### **4.3 YOLO tulemused**

Validatsiooni hulga puhul saavutas YOLO mudel kondensaatorite märkimisel 39% täpsuse. Tausta tuvastamisega sai mudel hakkama 100% ajast, mis tähendab et mudel oli pigem konservatiivne oma tuvastusega ning selle täpsust aitaks madalam tuvastus lävend. Kasutades mitmekordset käivitust ühe trükkplaadi vaatamisel on võimalik parandada tulemusi. Mudel on väga tundlik valgustingimuste muutustele, seega on mudeli täpsusele suureks abiks ühtlane valgustus. Jooksutades suudab mudel töötada reaalaajaliselt.

#### **4.4 FasterRCNN tulemused**

FasterRCNN mudel vajas kõvasti rohkem ettevalmistust. Pidin ümber tegema mudeli väljundkihi et see suudaks ennustada erikujul olevaid suunatud piirdkaste. Piirdkastid on määratud keskpunkti  $x$  ja  $y$  koordinaatidega, laiuks, pikkuse ja kraadidega. Käivitusel oli eelnevalt mainitud riistvara peal iga kaadri töötlemisaeg umbes sekund. Seega on tegu kõvasti aeglasema käivitusega, ning vähem võimsa riistvara puhul on see problemaatiline.

#### **4.5 Ülevaade praktilisest rakendusest**

Tulevikus saab luua praktiliseks lahenduseks GUI. Selle idee oleks kasutada etaloni põhise kontrolli, kus õigesti loodud trükkplaat antakse kaamerale ette, valitakse sobiv tuvastus, kus on ideaalis märgitud kõik vajalikud komponendid õigesti. Siis järgnevalt võrreldakse kõiki järgnevaid trükkplaate tuvastatud etaloniga. Kui järgneval plaadil on midagi tuvastusel väga erinev etalonist, siis rakendus annaks veast teada, ning seda saaks kontrollida töötaja.

## **5. Tulemuste analüüs**

### **5.1 Kaamera põhine kontroll**

Üldiselt on levinud tööstuses mitmekesisemad lahendused, kus ei kasutata ainult kaamera pilti vigade tuvastuseks. Kuigi sellised süsteemid on kõvasti sügavamad on nad suures osas ikkagi manuaalselt vaja üles seada iga erineva trükkplaadi kohta (IJECE 2019). Kaamera põhise kontrolli puhul on suurimateks eeliseks selle odavus ning lai üldistus võimalus. Kuna komponendid mida erinevatel trükkplaatidel tuvastada on sageli samasugused, siis on võimalik kontrollida ka eelnevalt tundmatuid plaate. Suurimaks probleemiks jäi minu lahenduse puhul vähene täpsus. Peamiselt on vähene tuvastustäpsus tingitud treeninghulga ja tuvastushulga suurest varieeruvusest. Ideeks oli luua valgustuselt ja positsiooniliselt varieeruvat treeninghulka, et treenitud mudelid oleks kasutatavad eri tingimuste all. Sellepärast jäi mudeli validatsiooni täpsus ka üsnagi kesiseks.

### **5.2 ResNeXt võrreldes YOLO mudeliga**

Võrrelda ResNeXt ja YOLO tulemusi olenevalt andmete kogusest ja nende täpsust peamiselt.

### **5.3 Mudelite nõudlikkus**

Võrdlen ResNeXt ja YOLO kiirust tuvastamisel ning märgin kas mõlemad on praktiliselt kasutatavad.

## **7. Tulemuste arutelu**

Järgnevad peatükid arutlevad, mida saaks tulevikus edasi arendada. Samuti toovad need välja mõningad probleemid, mille lahendamiseks oleks vaja erinevalt tulevikus läheneda, et edasised loodud lahendused oleksid täpsemad ja kasulikumad.

### **7.1 Märgendatud andmekogu**

Manuaalse märgendamise hulka tuleks tulevikus vähendada kasutades pool juhendatud õppimist. Selle kasutamine aitaks kaasa suure andmekogumi loomisel. Ideaalselt saaks suure osa andmekogumist nii valmis teha peale algse kogumi tegemist, ning kasutada seda edaspidi, et saavutada paremat täpsust mudelis. Veel üheks võimaluseks on kasutada ka *Zero-Shot* õppimist, sellega saaks vähendada tulevikus manuaalse märgendamise vajadust, kuid eeldatavasti selline lähenemine tõstaks ka riistvaralisi nõudeid kogu süsteemile. Kuna selle töö maht ei jõudnud praeguste mudelite abiga *Zero-Shot* ja pooljuhendatud õppe rakendamiseni jääb see tulevikku edasi arendamiseks. Selliste lahenduste rakendamiseks oleks ka vaja teha muudatus andmete märgenduskujule ja süsteemi struktuurile, süsteemil oleks võimekus võrrelda etaloni ja tuvastatavat trükkplaati eraldiseisvalt objektituvastusele.

### **7.2 Mudelite täpsus ning parandused mida rakendada tulevikus**

## Viidatud kirjandus

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
3. Wang C-Y; Yeh I-H; Liao, H-Y. YOLOv9: Learning What You Want to Learn Using Programmable Gradient Information. Arxiv 2024
4. Zhou, H; Jiang, F; Lu, H; SSDA-YOLO: Semi-supervised Domain Adaptive YOLO for Cross-Domain Object Detection. Arxiv 2022
5. Boonkong A; Khampitak K; Kaewfoongrunsi; Namkhun S; Hormdee D; Applying Deep Learning for Occluded Uterus and Fallopian Tube Detection for Laparoscopic Tubal Sterilization. IEEE 2024
6. Zhang, A; Lipton, Z; Li, M; Smola, A J; Dive into deep learning. Cambridge New York Port Melbourne New Delhi Singapore: Cambridge University Press. 2024
7. Redmon, J; Divvala S; Girshick R;Farhadi A; You Only Look Once: Unified, Real-Time Object Detection. Arxiv 2016
8. Larochelle, H; Erhan, D; Bengio Y; Zero-data Learning of New Tasks. Université de Montréal. IEEE 2008
9. Rehman, S; Fei, K; Lai N S; Automated PCB identification and defect detection system (APIDS). IJECE 2019
10. Girshick, R; Donahue, J; Darrell, T; Malik, J; Rich feature hierarchies for accurate object detection and semantic segmentation. Arxiv 2014

11. Andreu O. C; Smeraldi F; Semantic Image Cropping, Queen Mary University of London, Arxiv 2018
12. Y. Wang, Y. Zhang, Y. Zhang, L. Zhao, X. Sun and Z. Guo, SARD: Towards Scale-Aware Rotated Object Detection in Aerial Imagery, IEEE 2019