



Kauno technologijos universitetas

Panevėžio technologijų ir verslo fakultetas

3D spausdinimo broko vaizdo atpažinimo sistemos projektavimas ir tyrimas

Baigiamasis magistro studijų projektas

Laurynas Kvedaras

Projekto autorius

Lekt. dr. Olga Strikulienė

Vadovė

Panevėžys, 2022



Kauno technologijos universitetas

Panevėžio technologijų ir verslo fakultetas

3D spausdinimo broko vaizdo atpažinimo sistemos projektavimas ir tyrimas

Baigiamasis magistro studijų projektas
Valdymo technologijos (6211EX014)

Laurynas Kvedaras

Projekto autorius

Lekt. dr. Olga Strikulienė

Vadovė

Recenzentas / Recenzentė

Panevėžys, 2022



Kauno technologijos universitetas
Panevėžio technologijų ir verslo fakultetas
Laurynas Kvedaras

3D spausdinimo broko vaizdo atpažinimo sistemos projektavimas ir tyrimas

Akademinio sąžiningumo deklaracija

Patvirtinu, kad:

1. baigiamąjį projektą parengiau savarankiškai ir sąžiningai, nepažeisdama(s) kitų asmenų autoriaus ar kitų teisių, laikydamasi(s) Lietuvos Respublikos autorių teisių ir gretutinių teisių įstatymo nuostatų, Kauno technologijos universiteto (toliau – Universitetas) intelektinės nuosavybės valdymo ir perdavimo nuostatų bei Universiteto akademinės etikos kodekse nustatytų etikos reikalavimų;
2. baigiamajame projekte visi pateikti duomenys ir tyrimų rezultatai yra teisingi ir gauti teisėtai, nei viena šio baigiamojo projekto dalis nėra plagijuota nuo jokių spausdintinių ar elektroninių šaltinių, visos baigiamojo projekto tekste pateiktos citatos ir nuorodos yra nurodytos literatūros sąrašė;
3. įstatymų nenumatytų piniginių sumų už baigiamąjį projektą ar jo dalis niekam nesu mokėjęs(-usi);
4. suprantu, kad išaiškėjus nesąžiningumo ar kitų asmenų teisių pažeidimo faktui, man bus taikomos akademinės nuobaudos pagal Universitete galiojančią tvarką ir būsiu pašalinta(s) iš Universiteto, o baigiamasis projektas gali būti pateiktas Akademinės etikos ir procedūrų kontrolieriaus tarnybai nagrinėjant galimą akademinės etikos pažeidimą.

Laurynas Kvedaras

Patvirtinta elektroniniu būdu



Kauno technologijos universitetas
Panevėžio technologijų ir verslo fakultetas

TVIRTINU
TVKC vadovė
Doc. dr. Nida Kvedaraitė

Baigiamojo magistro projekto užduotis

Diplomantui **Laurynui Kvedarui**

Baigiamojo projekto tema (lietuvių kalba)	3D spausdinimo broko vaizdo atpažinimo sistemos projektavimas ir tyrimas
Baigiamojo projekto tema (anglų kalba)	Design and Research of an Image Recognition System for 3D Printing Defects

Patvirtinta 2022 m. balandžio 11 d. dekanu potvarkiu Nr. V25-13-10.

Parengto baigiamojo projekto įkėlimo į Moodle aplinką terminas iki 2022 m. birželio 1 d.

Duomenys, reikalavimai ir sąlygos baigiamajam projektui

Baigiamojo projekto užduotys / uždaviniai, kurie turi būti atskleisti projekte

1. Išnagrinėti 3D spausdinimo metodus.
2. Apibrėžti pagrindines 3D spausdinimo broko rūšis.
3. Išnagrinėti galimus 3D broko aptikimo metodus.
4. Išanalizuoti dirbtinio neuroninio tinklo apmokymo metodus ir jo panaudojimo 3D spausdinime galimybes.
5. Aprašyti pagrindinius vaizdo kameros kalibravimo būdus.
6. Suprojektuoti 3D spausdinimo broko vaizdo atpažinimo sistemą, parenkant fizinę ir programinę įrangą.
7. Sudaryti 3D spausdinimo broko vaizdo atpažinimo sistemos valdymo algoritmą ir aprašyti projekte naudojamus vaizdų apdorojimo metodus.
8. Ištirti suprojektuotą 3D spausdinimo broko vaizdo atpažinimo sistemą.

Vadovė lekt. dr. Olga Strikuliene
(vadovo pareigos, vardas, pavardė, parašas)

Užduotį gavau Laurynas Kvedaras
(studento vardas, pavardė, parašas)

2022 m. balandžio 19 d.

KVEDARAS, Laurynas. 3D spausdinimo broko vaizdo atpažinimo sistemos projektavimas ir tyrimas. Magistro studijų baigiamasis projektas / vadovė lekt. dr. Olga Strikulienė; Kauno technologijos universitetas, Panevėžio technologijų ir verslo fakultetas.

Studijų kryptis ir sritis: elektronikos inžinerija, technologijos mokslai.

Reikšminiai žodžiai: 3D spausdinimas, objektų atpažinimas, vaizdų apdorojimas.

Panevėžys, 2022. 63 p.

Santrauka

Šio darbo tikslas suprojektuoti ir ištirti 3D spausdinimo broko vaizdo atpažinimo sistemą, kuri gebėtų koreguoti 3D spausdintuvo parametrus priklausomai nuo rasto broko. Tikslui pasiekti būtina: išnagrinėti 3D spausdinimo ir broko aptikimo metodus, apibrėžti pagrindines 3D spausdinimo broko rūšis ir vaizdo kameros kalibravimo būdus, išanalizuoti dirbtinio neuroninio tinklo apmokymo metodus ir jo panaudojimo 3D spausdinime galimybes, sudaryti kuriamos sistemos valdymo algoritmą, parinkti fizinę ir programinę įrangą bei ištirti 3D spausdinimo broko vaizdo atpažinimo sistemą.

Pirmame šio projekto skyriuje aprašyti 3D spausdinimo metodai: liejant siūlę ant pagrindo, stereolitografika ir lazerinė miltelių sluoksnio sintezė bei paaiškintas kiekvieno metodo veikimo principas. Antrame skyriuje analizuoti dažniausiai pasitaikantys 3D spausdinimo defektai, iš kurių pasirinkti 5 pagrindiniai, naudoti objektams atpažinti: išsitempusi siūlė, objekto deformacija, išliejamas per didelis ar per mažas medžiagos kiekis, sluoksnių atsiskyrimas (skilinėjimas).

Trečiame skyriuje išnagrinėti broko aptikimo metodai, naudojant vaizdo atpažinimą su keliomis kameromis. Ketvirtame skyriuje išanalizuoti neuroninio tinklo mokymo būdai, iš kurių pasirinktas prižiūrimas mokymasis naudojamai aplikacijai. Penktame skyriuje aprašytas neuroninio tinklo panaudojimas 3D spausdinime, atliekant procedūrų stebėjimą, projektavimą ir taikant koreliaciją tarp proceso parametrų bei gauto komponento galutinių savybių. Šeštame skyriuje nagrinėti kameros kalibravimo būdai: automatinis kalibravimas, neplaninis kalibravimas ir naudojant šaškių lentą.

Septintame skyriuje aprašyta ir parinkta reikalinga įranga 3D spausdinimo broko vaizdo atpažinimo sistemai sukurti. Pasirinkti tokie defektų stebėjimo įrangos komponentai: mikrokompiuteris „Raspberry Pi“ 4b, kuriame kaupiamos nuotraukos ir palaikoma komunikacija su 3D spausdintuvu, „Raspberry Noir“ V2 8MP kamera tinkanti naudoti su mikrokompiuteriu. Parinkta programinė įranga: „Raspbian Debian“ 10, 32-bitų operacinė sistema, „Tensorflow“ 2.0.0 programa.

Aštuntame skyriuje atlikti bandymai su įvairių matmenų detalėmis, analizuoti atlikti veiksmai ir koreguotas parametrų valdymo algoritmas teisingiems sprendimams priimti. Sistema apmokyta aptikti 5 skirtingas broko rūšis. Bandymų metu sudaryti parametrų rinkiniai atkartojantys per didelio ar per mažo medžiagos kiekio broką. Taip pat atliktas algoritmo optimizavimas, kad broko tikrinimas būtų atliekamas atskirai kiekvienam defektui identifikuoti. Sukurtas nuotraukų kaupimo ir broko tikrinimo algoritmas, kuriuo naudojantis, tikrinama komunikacija su 3D spausdintuvu, apdorojamos nuotraukos ir priimami sprendimai defektams ištaisyti. Nuotraukų apdorojamos ir pažymimos 2 kadru per sekundę greičiu, kai nuotraukos rezoliucija 1280×720 pikselių. Naudojant spalvotas spausdinimo medžiagas, nustatyta, kad ryškiausiai matomos objekto savybės ir brokas naudojant žalios spalvos apšvietimą.

KVEDARAS, Laurynas. Design and Research of an Image Recognition System for 3D Printing Defects. Master's Final Degree Project / supervisor lect. dr. Olga Strikuliene; Panevėžys Faculty of Technologies and Business, Kaunas University of Technology.

Study field and area (study field group): Electronics Engineering, Technology Sciences (Engineering Sciences).

Keywords: 3D printing, object detection, image processing.

Panevėžys, 2022. 63 pages.

Summary

The aim of this work is to design and investigate a 3D printing defect image recognition system that would be able to adjust the 3D printer settings depending on the defect found. To achieve the goal it is necessary to: analyse 3D printing and defect detection methods, define the main types of 3D printing defects and video camera calibration methods, analyse the methods of artificial neural network training and its application in 3D printing, develop a control algorithm, select hardware and software for 3D printing defective image recognition system.

The first chapter of this project describes 3D printing methods: fused deposition modeling, stereolithography and laser powder bed fusion, and explains the principle of operation of each method. The second chapter analyzes the most common 3D printing defects, from which 5 main ones were selected: stringing, object deformation, over and under extrusion, layer separation (cracking).

The third section examines defect detection methods using multi-camera image recognition. In the fourth chapter, the neural network training methods are analyzed, from which supervised learning is selected for the application used. The fifth chapter describes the use of a neural network in 3D printing to monitor procedures, design, and apply correlations between process parameters and the final properties of the resulting component. In the sixth chapter, the methods of camera calibration are discussed: automatic calibration, non-planar calibration and using a checkerboard.

The seventh chapter describes and selects the necessary equipment to create a 3D printing defect image recognition system. The following defect monitoring equipment components have been selected: "Raspberry Pi" 4b microcomputer, which stores photos and supports communication with a 3D printer, "Raspberry Noir" V2 8MP camera suitable for use with a microcomputer. Selected software: "Raspbian Debian" 10, 32-bit operating system, "Tensorflow" 2.0.0.

In the eighth chapter, tests with 3D printed objects of various dimensions were performed, the decisions made were analyzed, and the parameter control algorithm for making the right decisions was adjusted. The system is trained to detect 5 different types of defects. During the tests, parameter sets were created that replicate the defect of over or under extrusion. Algorithm optimization was performed to identify each defect separately. An algorithm for photo collection and defect inspection has been developed, which is used to check communication with the 3D printer, process photos and make decisions to correct defects. Photos are processed and tagged at a rate of 2 frames per second at a photo resolution of 1280×720 pixels. When using colorful filament, the subject properties and defects are found to be most visible when using green lighting.

Turinys

Lentelių sąrašas.....	9
Paveikslų sąrašas	10
Santrumpų sąrašas	12
Įvadas.....	13
1. 3D spausdinimo metodai.....	14
1.1. Sulydytos medžiagos išspaudimas ant pagrindo	14
1.2. Stereolitografija	14
1.3. Lazerinė miltelių sluoksnio sintezė	15
2. 3D spausdinimo brokas.....	17
2.1. Nesuderintas pagrindo plokštės aukštis.....	17
2.2. Medžiagos srauto sutrikimas ar ekstruderio užsikimšimas	17
2.3. Neteisingai nustatyti spausdintuvo parametrai	18
2.4. Išliejamas per didelis ar per mažas medžiagos kiekis	19
2.5. Sluoksnių atsiskyrimas (sutrūkinėjimas).....	20
3. Broko aptikimo metodai	21
3.1. Vaizdo kamera.....	21
4. Dirbtiniai neuroniniai tinklai	23
4.1. Neuroninio tinklo mokymasis	23
4.1.1. Prižiūrimas mokymasis	23
4.2. Neprižiūrimas mokymasis	25
4.2.1. Sustiprintas mokymasis	25
4.3. Konvoliucinis neuroninis tinklas	25
5. Dirbtinio neuroninio tinklo panaudojimas 3D spausdinime	27
5.1. Procesos stebėjimas.....	27
6. Kameros kalibravimas	28
6.1. Kalibravimo metodai	28
6.1.1. Automatinis kalibravimas.....	28
6.1.2. Neplaninio objekto kalibravimas.....	29
6.1.3. Kalibravimas naudojant šaškių lentą	29
7. Eksperimentinė dalis	30
7.1. Kuriamos ir rinkoje esančios sistemų palyginimas	30
7.2. Fizinė įranga	30
7.2.1. Mikrokompiuteris „Raspberry Pi“ 4b.....	31
7.2.2. Kameros parinkimas	31
7.2.3. Šviesos šaltinis.....	32
7.2.4. Įrangos laikikliai.....	33
7.3. Programinė įranga	34
7.3.1. 32-bitų operacinė sistema „Raspbian Debian“ 10	35
7.3.2. „TensorFlow“ 2.0.0 biblioteka	35
7.3.3. „Ultimaker Cura“ 4.6 programa	35
7.4. Nuotraukų kaupimas.....	36
7.4.1. Žodinis algoritmas nuotraukoms kaupti	37
7.4.2. Algoritmas nuotraukoms kaupti	37

7.4.3. Nuotraukos neuroninio tinklo mokymui	38
7.5. Neuroninio tinklo apmokymas	41
7.6. Apmokyto modelio naudojimas	42
7.7. Broko identifikavimas naudojantis „Google Colab“	43
7.8. Parametrų valdymas	45
7.8.1. Žodinis valdymo algoritmas	45
7.8.2. Struktūrinis valdymo algoritmas	47
8. Tiriamoji dalis.....	49
8.1. „Raspberry Pi“ mikrokompiuterio centrinio procesoriaus temperatūros valdymas	49
8.2. 3D spausdintuvo parametrų valdymas.....	49
8.2.1. 1 bandymas.....	50
8.2.2. 2 bandymas.....	50
8.2.3. 3 bandymas.....	52
8.2.4. 4 bandymas.....	54
8.3. Algoritmo optimizavimas.....	55
8.4. Spalvoto apšvietimo bandymas	56
Išvados	60
Literatūros sąrašas	61

Lentelių sąrašas

1 lentelė. „Raspberry Pi“ 4b specifikacija [24]	31
2 lentelė. „Raspberry Noir“ V2 ir „Logitech“ C920E palyginimas [25, 26]	31
3 lentelė. Objektų paviršių nuotraukos apšviečiant skirtingos spalvos šviesa (1)	56
4 lentelė. Objektų paviršių nuotraukos apšviečiant skirtingos spalvos šviesa (2)	58

Paveikslų sąrašas

1 pav. FDM spausdinimo technologija [1].....	14
2 pav. SLA spausdinimo technologija [2].....	15
3 pav. LPBF spausdinimo technologija [4].....	16
4 pav. Gaminys neprilipęs prie pagrindo [6].....	17
5 pav. Užsikišęs ekstruderis	18
6 pav. Išsitempusi siūlė	18
7 pav. Per didelis išliejamos medžiagos kiekis	19
8 pav. Per mažas išliejamos medžiagos kiekis.....	19
9 pav. Sluoksnių atsiskyrimas	20
10 pav. Kamerų išdėstymas aplink spausdintuvą [9] (trys skirtingi CAD modelio vaizdai, vaizduojantys standus, spausdintuvą ir jų padėtį): a) vaizdas iš viršaus; b) vaizdas iš kairiojo kampo; c) vaizdas iš priekio	21
11 pav. Skirtingo ryškumo ribos: a) pilnas objektas; b) nebaigtas spausdinti; c) rodomas skirtumas tarp dalinai ir pilnai atspausdintų objektų; d) ryškumo riba 50; e) ryškumo riba 75; f) ryškumo riba 100 [9].....	22
12 pav. Aktyvavimo funkcijos [10]	23
13 pav. Apmokomo tinklo modelis [12]	24
14 pav. Prižiūravimo tinklo mokymosi ir naudojimo schema [14]	24
15 pav. Neprižiūravimo tinklo mokymosi schema [15].....	25
16 pav. Konvoliucinis neuroninis tinklas [18]	26
17 pav. Konvoliucinio neuroninio tinklo algoritmas [19].....	27
18 pav. RANSAC algoritmo vizualizacija [21]	28
19 pav. „Žuvies akies“ efektas [22]: a) prieš kalibravimą; b) po kalibravimo	29
20 pav. Aptikti spausdinimo defektai [23] (nustatytas galimas gedimas – angl. „ <i>possible failure detected</i> “; klaida – angl. „ <i>failing</i> “; tinkamas – angl. „ <i>looking good</i> “).....	30
21 pav. 1 bandymo apšviestas plotas	32
22 pav. 2 bandymo apšviestas plotas	33
23 pav. 3 bandymo apšviestas plotas	33
24 pav. Modifikuotas laikiklis.....	34
25 pav. Kameros ir apšvietimo laikiklis.....	34
26 pav. Modelio peržiūra	36
27 pav. „Ultimaker Cura“ papildiniai	36
28 pav. Nuotraukų kaupimo algoritmas	37
29 pav. Nuotraukų kaupimo programinis kodas	38
30 pav. Nuotraukų apkarpymas.....	39
31 pav. Palyginamos nuotraukos prieš apdorojimą (a, c, e) ir po apdorojimo (b, d, f).....	40
32 pav. Palyginamos nuotraukos prieš apdorojimą (a, d) ir po apdorojimo (b, e).....	40
33 pav. Tinklo apmokymui naudojamas programinis kodas	41
34 pav. Tinklo mokymasis	42
35 pav. Programinis kodas objekto defektams nustatyti.....	42
36 pav. Atpažintas brokas nuotraukose: a, d – išsitempusi siūlė; b, c – per didelis išliejamos medžiagos kiekis.....	43
37 pav. Apmokomo tinklo aplankų išdėstymas	44
38 pav. Defektų identifikavimo aprašo kodas.....	44

39 pav. Gautas rezultatas apmokant tinklą iki 1000 žingsnių.....	45
40 pav. Defektų identifikavimas	45
41 pav. Struktūrinis valdymo algoritmas (1).....	47
42 pav. Struktūrinis valdymo algoritmas (2).....	48
43 pav. „RaspberryPi“ CPU temperatūra su ir be aušinimo	49
44 pav. 1 bandymo rezultatas.....	50
45 pav. 2 bandymo rezultatas.....	51
46 pav. Bandymo metu sukaupti duomenys	51
47 pav. Kaitinamo ekstruderio temperatūros pokytis priimant sprendimus (1).....	52
48 pav. Objekto paviršius keičiantis temperatūrai	52
49 pav. 3 bandymo rezultatas.....	53
50 pav. Kaitinamo ekstruderio temperatūros pokytis priimant sprendimus (2).....	53
51 pav. Nustatytas neteisingas defektas	54
52 pav. Neatpažintas deformacijos brokas.....	54
53 pav. Nustatytas deformacijos brokas.....	55
54 pav. Vizualizuojami aptikti defektai	55

Santrumpų sąrašas

Santrumpos:

FDM – spausdinimo metodas liejant siūlę ant pagrindo (angl. *fused deposition modeling*);

SLA, STL – stereolitografika (angl. *stereolithography*);

LPBF – lazerinė miltelių sluoksnio sintezė (angl. *laser powder bed fusion*);

ANN – dirbtinis neuroninis tinklas (angl. *artificial neural network*);

CNN – konvoliucinis neuroninis tinklas (angl. *convolution neural network*);

TPU – termoplastinis poliuretanas (angl. *thermoplastic polyurethane*);

TPE – termoplastinis elastomeras (angl. *thermoplastic elastomer*);

CAD – kompiuterinis dizainas (angl. *computer-aided design*);

3D – trimatis (angl. *three-dimensional*);

ML – mašininis mokymasis (angl. *machine learning*);

AM – gamyba, naudojant medžiagos pridėjimą (angl. *additive manufacturing*);

RANSAC – atsitiktinės imties vienodumas (angl. *random sample consensus*);

CAM – kompiuterinis apdirbimas (angl. *computer-aided manufacturing*);

CAE – kompiuterinė gamyba (angl. *computer-aided engineering*);

PCB – spausdinta elektrinė plokštė (angl. *printed circuit board*);

CPU – centrinis procesoriaus blokas (angl. *central processing unit*);

GPU – grafinis procesoriaus blokas (angl. *graphics processing unit*).

Įvadas

3D spausdinimas gali būti viena iš pagrindinių technologijų ateities gamybos sistemoms. Šios spausdinimo sistemos turi specifinių pažeidžiamumų, kurie gali paveikti gaminio vidinius sluoksnius, o išorėje atrodyti kokybiškai. Spausdinimo procesas ne visada yra akylai prižiūrimas operatorių, todėl įvykus nenumatytiems pasikeitimams, dėl kurių gaminys gali būti spausdinamas neteisingai, gali tęstis toliau. Vienas iš populiariausių metodų tokiems defektams kaip: sluoksnių įtrūkimai, liejamas perteklius, užsikišęs ekstruderis ir kita, aptikti – vaizdo atpažinimo įrangos naudojimas.

Tyrimo objektas – 3D spausdinimo broko vaizdo atpažinimo sistema.

Projekto tikslas: suprojektuoti ir ištirti 3D spausdinimo broko vaizdo atpažinimo sistemą.

Uždaviniai:

1. Išnagrinėti 3D spausdinimo metodus.
2. Apibrėžti pagrindines 3D spausdinimo broko rūšis.
3. Išnagrinėti galimus 3D broko aptikimo metodus.
4. Išanalizuoti dirbtinio neuroninio tinklo apmokymo metodus ir jo panaudojimo 3D spausdinime galimybes.
5. Aprašyti pagrindinius vaizdo kameros kalibravimo būdus.
6. Suprojektuoti 3D spausdinimo broko vaizdo atpažinimo sistemą, parenkant fizinę ir programinę įrangą.
7. Sudaryti 3D spausdinimo broko vaizdo atpažinimo sistemos valdymo algoritmą ir aprašyti projekte naudojamus vaizdų apdorojimo metodus.
8. Ištirti suprojektuotą 3D spausdinimo broko vaizdo atpažinimo sistemą.

Tyrimo metodai: mokslinės literatūros analizė, eksperimentai.

Autoriaus leidybai pateiktas straipsnis

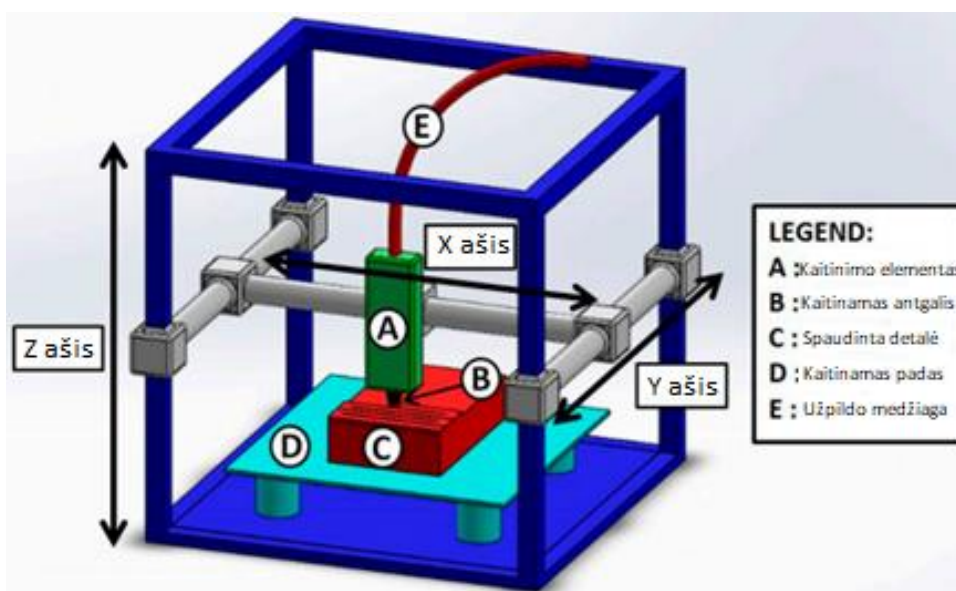
KVEDARAS, Laurynas; STRIKULIENĖ, Olga. 3D spausdinimo broko vaizdo atpažinimo sistemos projektavimas ir tyrimas // Technologijų ir verslo aktualijos – 2022: studentų mokslinių darbų konferencijos pranešimų medžiaga, Lietuva, Panevėžys, 2022 m. lapkričio 25 d. / Kauno technologijos universiteto Panevėžio technologijų ir verslo fakultetas. Kaunas: Kauno technologijos universitetas. ISSN 2538-8045. 2022.

1. 3D spausdinimo metodai

3D spausdinimui naudojamos medžiagos parenkamos, atsižvelgiant į prototipų kūrimui naudojamos technologijos. Kai kuriose technologijose naudojamos šviesai jautrios dervos, sukietintos lazeriu arba DLP projektoriaus šviesa (angl. *polyJet and stereolithography*, SLA). Taikant spausdinimo metodą, kuriame siūlė liejama ant pagrindo, galima pagaminti objekto dalis iš įvairių medžiagų, įskaitant elastomerus [1].

1.1. Sulydytos medžiagos išspaudimas ant pagrindo

Sulydytos medžiagos išspaudimo ant pagrindo technologija (angl. *fused deposition modeling*, FDM) – tai 3D spausdinimo technologija (žr. 1 pav.), veikianti „priedo“ principu, kai iš ritės vyniojama plastiko gija šildoma iki plastiko lydymosi temperatūros, liejama sluoksniais [2].

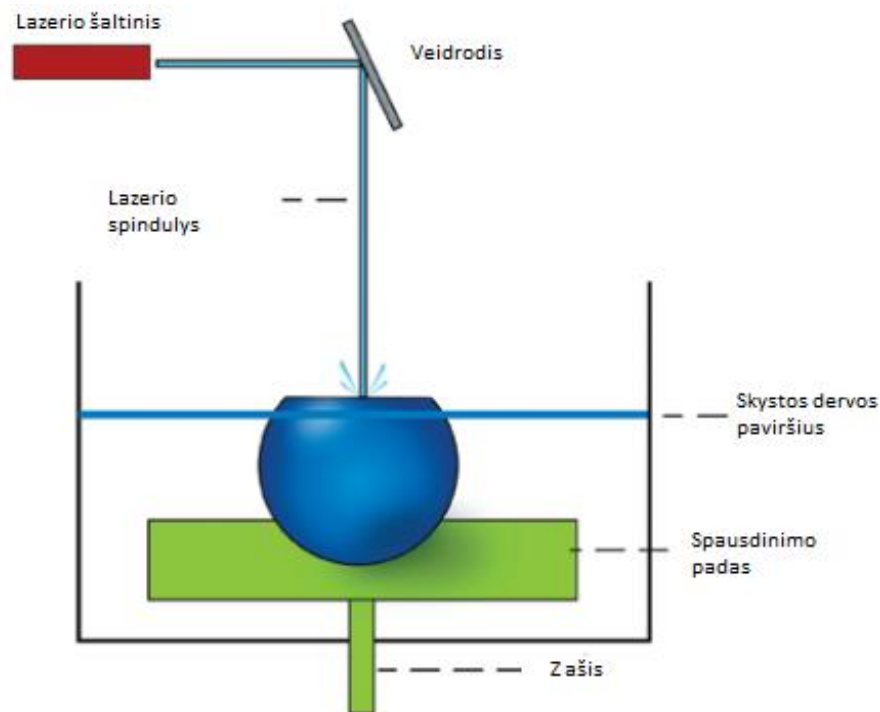


1 pav. FDM spausdinimo technologija [1]

FDM technologijai reikalingos programinės įrangos, kurios gebėtų apdoroti stereolitografinius failus ir paversti 3D objekto modelį į spausdintuvui suprantamą spausdinimo formatą. Jei reikia, gali būti sukurtos atraminės konstrukcijos. Objektas gaminamas liejant termoplastinę medžiagą, kuri sukietėja nepraėjus nei sekundei po išpurškimo. Spausdintuvo purkštuką galima judinti tiek horizontaliai, tiek vertikaliai, valdant jį kompiuterinės gamybos (CAM) programinės įrangos paketu. Suprojektuota dalis spausdinama iš apačios į viršų, po vieną sluoksnį. Žingsniniai varikliai naudojami ašiai judinti taip, kad liejant medžiagą purkštukas būtų nustatytoje koordinatėse [2].

1.2. Stereolitografija

Stereolitografija (SLA) yra dar viena 3D technologijos rūšis (žr. 2 pav.), naudojama audinių inžinerijos srityje. Šiuo atveju objekto gamyba pagrįsta skystos dervos sukietėjimu fotopolimerizacijos būdu. Lazeris, sutelktas į tam tikrą gylį dervos talpykloje, sukelia lokalizuotą polimerizaciją (sukietėjimą). Sukietėjus sluoksniui, padėklas nusileidžia atstumu, lygiu sluoksnio storiui, nesukietėjusiai dervai padengiant ankstesnį sluoksnį. Šis procesas kartojamas tiek kartų, kol spausdinamas objektas įgauna formą [3].



2 pav. SLA spausdinimo technologija [2]

Spausdinant SLA metodu, lazerio skleidžiamai energijai įtaką daro šviesos šaltinio galia, nuskaitymo greitis, ekspozicijos laikas ir polimero bei fotoiniciatoriaus kiekiai. SLA spausdintuvai paprastai yra lėtesni nei kiti, nes lazeris vienu metu gali apšviesti tik mažą plotą, o taip pat reikalingas švitinimas ultravioletinėje šviesoje.

SLA technologijos skiriamoji geba 10 kartų mažesnė ($20\ \mu\text{m}$), lyginant su kitomis gamybos technologijomis ($50\text{--}200\ \mu\text{m}$) [2, 3].

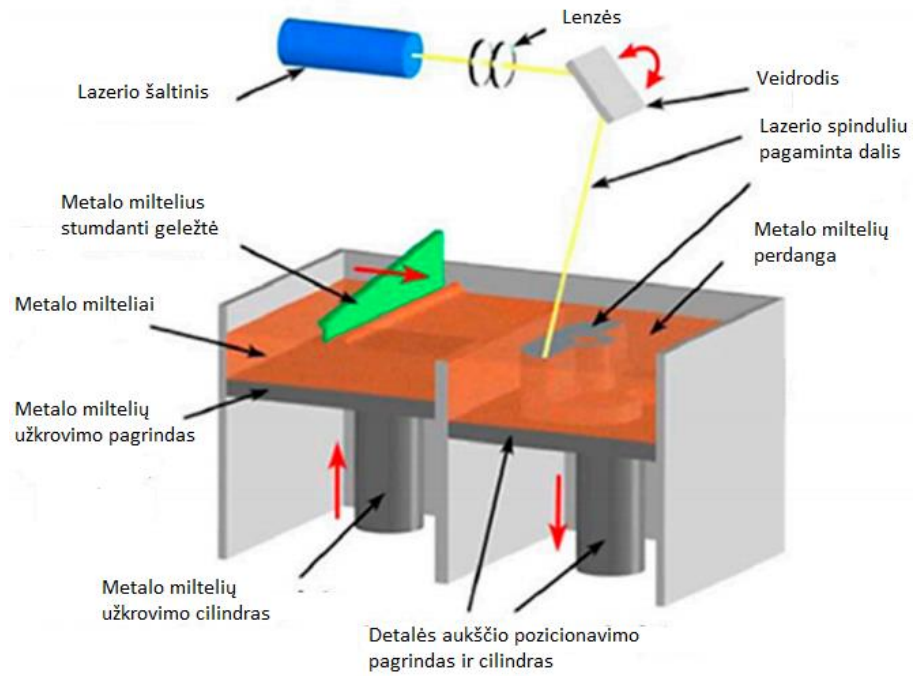
1.3. Lazerinė miltelių sluoksnio sintezė

Lazerinė miltelių sluoksnio sintezė (angl. *laser powder bed fusion*, LPBF) – tai dažniausiai naudojama metalo AM technologija visame pasaulyje.

Metalo milteliai suberiami į tuščią kamerą, kurioje objektai gaminami sluoksniuotai, naudojant didelės galios lazerį (paprastai vienos nuolatinės bangos pluošto lazeris, veikiantis $1075\ \text{nm}$ bangos ilgiu), kuris selektyviai sulydo miltelius pagal pateiktą kompiuterinio projektavimo (CAD) modelį. Proceso metu (žr. 3 pav.), metalo milteliai iš tiekimo sistemos paskleidžiami stumiant geležtę [4].

Pagrindas, sumontuotas ant cilindro, nuleidžiamas į tokį pat aukštį, kaip ir tirpinamas metalo miltelių storis. Proceso kartojamas sluoksnis po sluoksnio, kol gaunamas galutinis objektas. Atliekant sintezę būtinas nuolatinis inertinių dujų (argono ar azoto) srautas [4].

Miltelių sluoksnis apibūdinamas, kaip sluoksnio tankis (priklauso nuo miltelių paskleidimo ant objekto plokštumos ir tarp jo neužpildytų sričių), veikiamas lazerio. Lazeriu ištirpinamos padengtos milteliais pagrindo sritys.



3 pav. LPBF spausdinimo technologija [4]

Likę neapdoroti milteliai panaudojami, kaip papildoma gaminamo objekto atrama [4].

2. 3D spausdinimo brokas

Spausdinant FDM spausdintuvu, neprižiūrimas gaminys neretai gaunamas brokuotas. Tai dažniausiai nutinka nutraukus medžiagos srautą, o tai lemia mažų ir didelių gaminio defektų atsiradimą. Mažų nukrypimų galima nevertinti, jei ankstesni arba paskutiniai sluoksniai kompensuoja trūkstamą ar sugadintą sluoksnį. Priešingai, jeigu atsiranda didelių defektų, 3D spausdinta dalis negali atlikti norimos funkcijos, nes atsiranda formos nukrypimų bei struktūros dalių susilpnėjimų [5]. Dažniausiai pasitaikantys spausdinimo defektai:

- nesuderintas pagrindo plokštės aukštis;
- užsikišęs ekstruderis;
- medžiagos srauto sutrikimas ar pasibaigusi medžiaga;
- sukibimo su pagrindo plokšte trūkumas;
- neteisingai nustatyti spausdintuvo parametrai;
- išliejamas per didelis ar per mažas medžiagos kiekis;
- sluoksnių atsiskyrimas (skilinėjimas).

2.1. Nesuderintas pagrindo plokštės aukštis

Vienas svarbiausių valdymo parametru, siekiant užtikrinti aukštą spausdinimo kokybę – purkštuko aukštis, skirtas pirmam sluoksniui spausdinti. Jei aukštis per didelis – spausdinta dalis gali neprilipti prie spausdinimo pagrindo [6]. 4 paveiksle matomas sluoksnio pasistūmimas, kuris įvyksta dėl netinkamo purkštuko aukščio. Spausdinant pirmąjį sluoksnį derinimas vyksta rankiniu būdu. Jei atstumas mažesnis nei 200 μm, plastiko kiekis gali išsilieti į plotį, dėl to praplatėja siūlė, arba užstrigti ekstruderyje ir apriboti plastiko tiekimą.



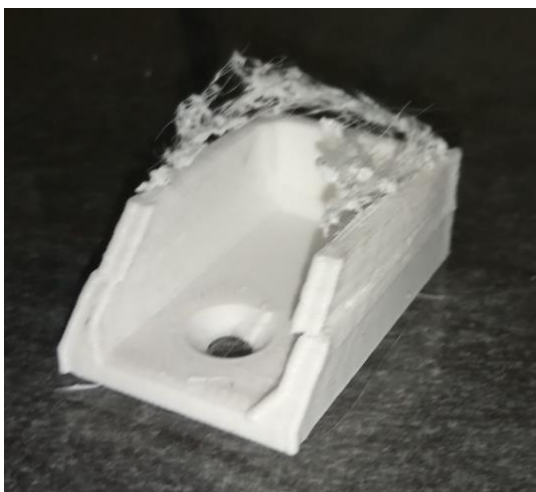
4 pav. Gaminys neprilipęs prie pagrindo [6]

Sustingęs perteklius gali trukdyti ekstruderiiui judėti, todėl užkliudžius detalę, ji bus nustumta nuo pagrindo. Priešingai, jeigu atstumas didesnis nei nustatytas sluoksnio storis (~200 μm), liejamas plastikas neprilips prie pagrindo.

2.2. Medžiagos srauto sutrikimas ar ekstruderio užsikišimas

Esant temperatūrų kaitoms nuo 20 °C prieš spausdinimą ir 200 °C po spausdinimo, plastikas šildomame ekstruderyje sukietėja arba suminkštėja. Nuo vidinės ekstruderio sienos gali atkibti karščio paveiktas plastikas, kuris nevysiškai išsilydo ir taip užkemša ekstruderį. Taip pat, ši problema dažnai matoma spausdinant plastiką su įvairiomis priemaišomis, tokiomis kaip, sutvirtinančios

plokštelės, trumpi pluoštai, įskaitant susmulkintus anglies pluoštus, polimero skaidulas, anglies nanovamzdelius ir stiklo pluoštus [7]. Užsikišus ekstruderiiui, plastikas neprateka, tačiau užduotos komandos toliau vykdomos, nepriklausomai nuo to ar plastikas liejamas ant sugadinto ar nesamo sluoksnio. Pavyzdys pateiktas 5 paveiksle.

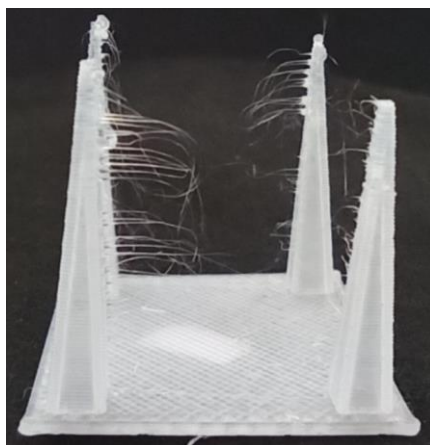


5 pav. Užsikišęs ekstruderis

Spausdinant iš minkštesnių medžiagų (TPU, TPE), kurios pradeda lydintis žemesnėje, t. y. 145 °C, temperatūroje ir yra lanksčios, pasitaiko, kad esant didesniai nei 50 mm/s srautui, plastikas nespėja pratekėti per ekstruderį, o spausdinimo gija įtraukiama į mechanizmą stumiantį plastiko siūlę.

2.3. Neteisingai nustatyti spausdintuvo parametrai

Spausdintuvo nustatymai leidžia valdyti tokius kintamuosius, kaip sluoksnio storis, atramos kampas, lydymosi temperatūra, pagrindo temperatūra, spausdinimo greitis, plastiko išstūmimo srauto santykis, ekstruderio atstumas, užpildymo tipas, užpildymo tankis, paviršiaus sluoksniai, pagalbines atramos, siūlės tipas ir ventiliatoriaus greitis. Visi šie parametrai yra svarbūs spausdinimo kokybei [5]. Dažniausia problema, susijusi su spausdintuvo nustatymais – siūlės ištempimas pereinant nuo vieno paviršiaus prie kito (žr. 6 pav.).

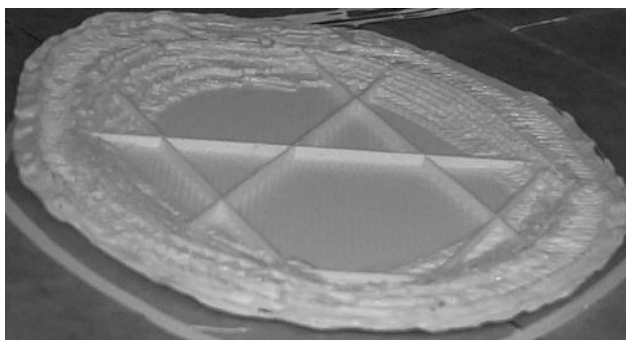


6 pav. Išsitempusi siūlė

Kiekviename sluoksnio perėjime, užsilikęs medžiagos perteklius nespėja atvėsti, todėl prilimpa prie kaitinamo ekstruderio ir ištempiamas judesio metu.

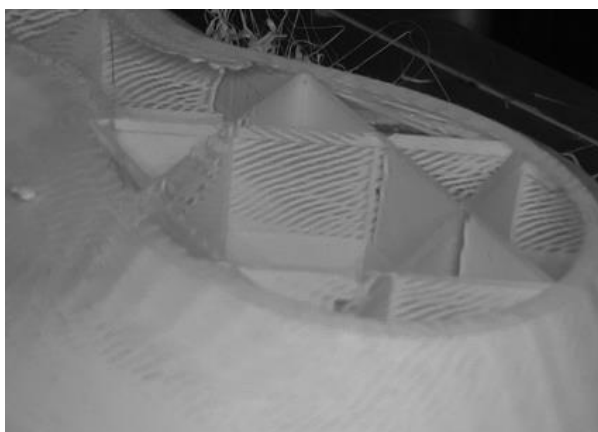
2.4. Išliejamas per didelis ar per mažas medžiagos kiekis

Per didelis išspaudžiamos medžiagos kiekis matomas tuomet, kai sluoksniai nėra tolygūs (žr. 7 pav.), t. y. vietoje vientiso išliejimo gaunamas gumbuotas ar per aukštas sluoksnis. Dėl to įkaitintas ekstruderis pradeda siekti prieš tai esantį sluoksnį, dėl to perteklinė medžiaga gali aplipti aplink patį ekstruderį. Medžiagos kiekį galima kontroliuoti per kiekio daugiklio parametą, kuris dažniausiai būna 100 proc. Jeigu temperatūra per didelė naudojamam plastikui lydyti, sunkiau kontroliuojamas savaeigis tekėjimas iš ekstruderio.



7 pav. Per didelis išliejamos medžiagos kiekis

Per mažas medžiagos kiekis pavaizduotas kitame paveiksle (žr. 8 pav.). Per mažas išspaudžiamos medžiagos kiekis lemia oro tarpų tarp sluoksnių susidarymą, sluoksnių nesusiliejamą vienas su kitu. Taip pat pastebima, kad paviršius tampa ruplėtas, vietomis peršviečiamas, o objekto išorinis sluoksnis silpnas ir nesunkiai gniuždomas.



8 pav. Per mažas išliejamos medžiagos kiekis

Pagrindiniai parametrai: medžiagos kiekio reguliavimas ir kaitinamo ekstruderio temperatūra. Ši broko rūšis gali pasireikšti ir esant daliniam ekstruderio užsikimšimui.

2.5. Sluoksnių atsiskyrimas (sutrūkinėjimas)

Išorinių sluoksnių skilinėjimas arba viso sluoksnio nesukibimas susidaro, kai spausdinimas vyksta per dideliu atstumu nuo prieš tai buvusio sluoksnio, tokiu atveju plastiko gija nesulimpa viena su kita, dėl to atvėstant medžiagai susidaro paviršiaus įtempimas ir sluoksniai atsiskiria (žr. 9 pav.).



9 pav. Sluoksnių atsiskyrimas

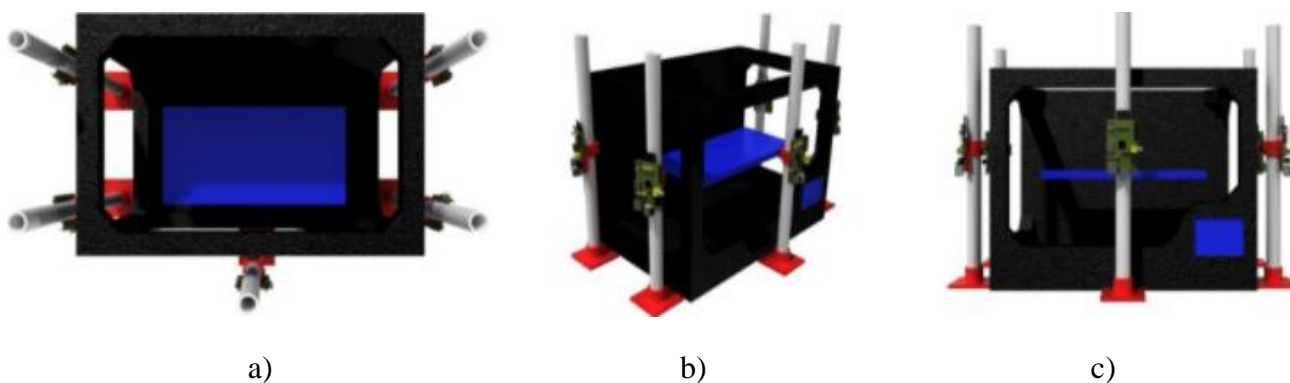
Taip pat sluoksnių atsiskyrimo savybė matoma ABS plastikuose, kuriems reikalinga ne tik didesnė ekstruderio temperatūra (apie 240 °C), bet ir oro temperatūra (apie 40 °C). Palaikant nurodytas temperatūras sumažinamas laikas, per kurį išspaudžiama medžiaga atvėsta. Pasibaigus spausdinimui, objektas jau būna sutvirtėjęs, todėl sumažėja tikimybė sluoksniams atsiskirti.

3. Broko aptikimo metodai

Daugelis 3D spausdintuvų, neturi pritaikytos sistemos spausdinimo proceso eigai stebėti. 3D spausdintuvai gali tęsti dalies spausdinimą tol, kol visi sluoksniai bus baigti, net jei baigėsi medžiagos ar yra galimų spausdinimo defektų, kurių nustatymas atliekant kokybės patikrinimą įvairiais (kritiniais) spausdinimo proceso etapais ne tik padeda užtikrinti taisomąsias priemones, bet ir eliminuoti medžiagų ir laiko švaistymą [8].

3.1. Vaizdo kamera

Vaizdiniams gedimams aptikti, 3D spausdinimo saugumui užtikrinti ir spausdinimo procesui stebėti gali būti naudojamos kameros sumontuotos skirtinguose įrenginio kampuose [5]. Pagamintų dalių kokybei įvertinti gali būti panaudota daugiakamerė sistema, susidedanti iš penkių kamerų ir vaizdo apdorojimo programinės įrangos, sukurtos naudojant „C#“ ir „.NET framework“ sistemas. Nebaigto objekto vaizdai fotografuojami aštuoniose vietose (žr. 10 pav.) iš penkių kamerų, pastatytų penkiais skirtingais kampais (iš viso padaroma 40 nuotraukų) ir palyginami su galutiniu objekto vaizdu [9].



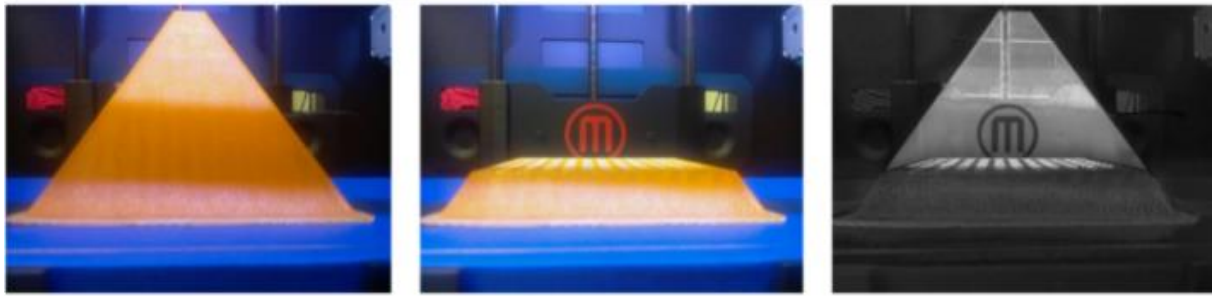
10 pav. Kameros išdėstymas aplink spausdintuvą [9] (trys skirtingi CAD modelio vaizdai, vaizduojantys standus, spausdintuvą ir jų padėtį): a) vaizdas iš viršaus; b) vaizdas iš kairiojo kampo; c) vaizdas iš priekio

Naudojant šį metodą galima aptikti dviejų tipų klaidas: kai spausdinimas sustabdytas progreso viduryje, dėl kurio objektas buvo nebaigtas, ir kai dėl spausdintuvo problemos nepavyksta išlieti plastiko iš kaitinamo ekstruderio [9]. Pabaigto ir sustabdyto spausdinimo metu objektų nuotraukos lyginamos pikseliais.

Šio palyginimo rezultatas yra abiejų vaizdų skirtumų nustatymas. 11 paveiksle parodytas vaizdo skirtumo lygis: ryškiausi plotai rodo didžiausią skirtumo lygį.

Šis vaizdas sukuriamas kiekvienam pikseliui priskiriant ryškumo (raudonos, žalios ir mėlynos spalvos) reikšmes [9]. Rodomi pikseliai, kurių ryškumo ribos yra 50, 75 ir 100. Šiose nuotraukose juodos sritys yra reikšmingos, o baltos – nepaisomos.

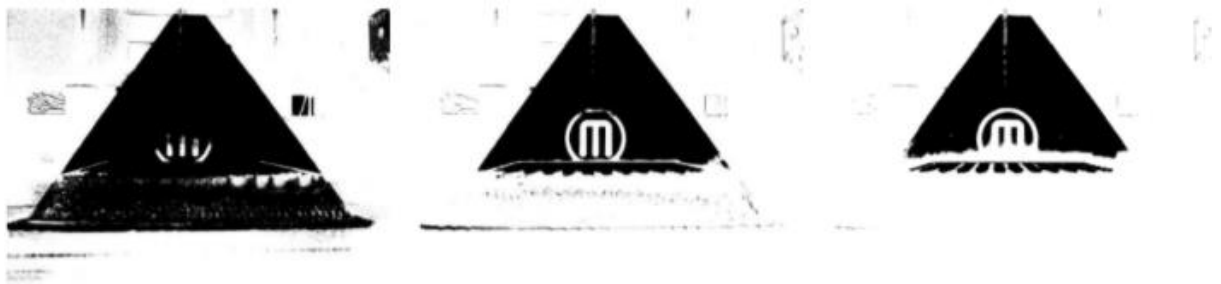
Esant brokui, kai detalė atlimpa nuo pagrindo, balta nuotraukos dalis pasislinks, taip išryškindama juodos formos sritį.



a)

b)

c)



d)

e)

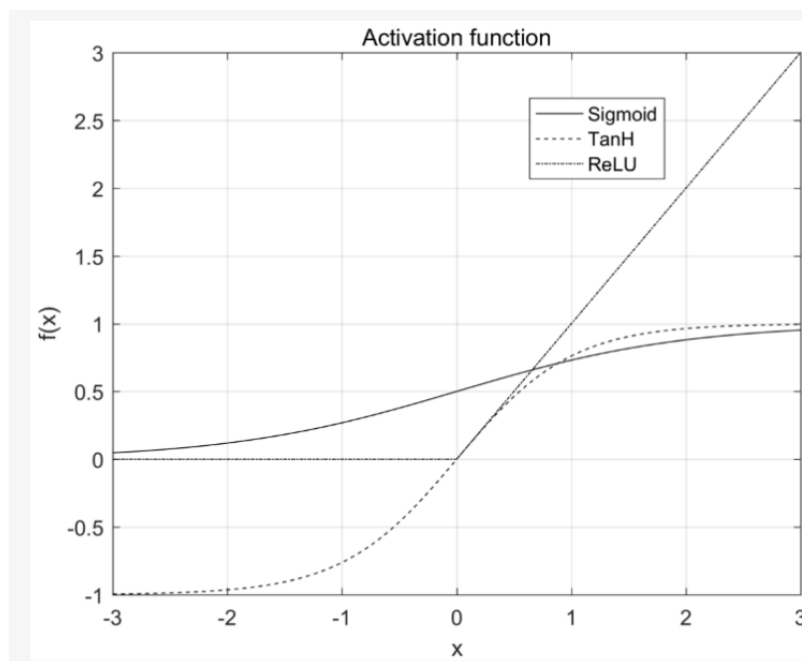
f)

11 pav. Skirtingo ryškumo ribos: a) pilnas objektas; b) nebaigtas spausdinti; c) rodomas skirtumas tarp dalinai ir pilnai atspausdintų objektų; d) ryškumo riba 50; e) ryškumo riba 75; f) ryškumo riba 100 [9]

Tokiu atveju, lyginant nuotraukas matoma nesutaptis ir spausdinimas stabdomas.

4. Dirbtiniai neuroniniai tinklai

ANN yra tarpusavyje sujungta mazgų grupė. Mazgas, kuris priima įvesties signalą, jį apdoroja ir perduoda išėjimo signalą į savo kaimyninius neuronus, vadinamas dirbtiniu neuronu. Signalo apdorojimas, dar vadinamas sklidimo, transformavimo, perdavimo funkcija arba aktyvavimo funkcija, yra matematiškai įgyvendinamas. 12 paveiksle pavaizduotos aktyvinimo funkcijos: sigmoidinė (angl. *sigmoid*), hiperbolinė (*TanH*) ir linijinio lyginimo vieneto (*ReLU*) [10].



12 pav. Aktyvavimo funkcijos [10]

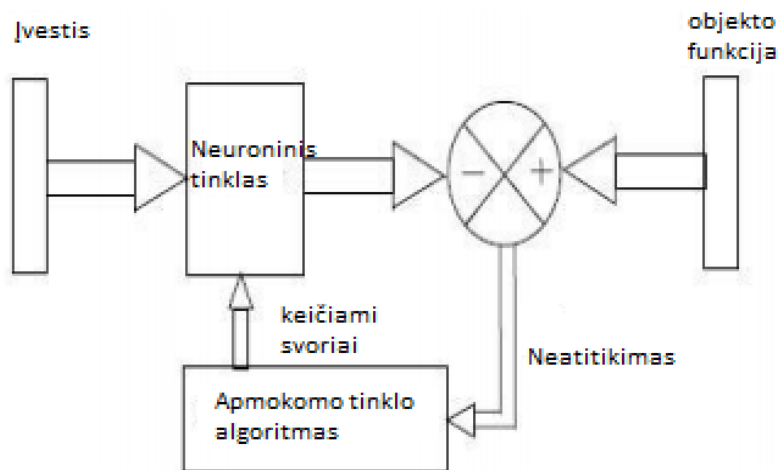
Dirbtinių neuroninių tinklų atsiradimą paskatino ankstyvieji jutiminiai smegenų apdorojimo modeliai. Dirbtinį neuronų tinklą galima sukurti imituojant modelio neuronų tinklą kompiuteryje. Taikant algoritmus, imituojančius realių neuronų procesus, galima priversti tinklą „mokyti“ spręsti daugelio tipų problemas [11].

4.1. Neuroninio tinklo mokymasis

Dirbtiniuose neuroniniuose tinkluose mokymasis vyksta, kai yra keičiami svoriai esantys tarp tam tikrų tinklo mazgų. Mokymasis – tai procesas, kai atsitiktinai vertinami neuroninio tinklo parametrai (svoriai), kurie yra nepertraukiami viso modeliavimo proceso metu ir pritaikomi aplinkoje, kurioje yra įdėtas neuroninis tinklas. Mokymosi greitis apibrėžiamas kaip greitis, kuriuo tinklas prisitaiko prie išorinių veiksnių. Mokymosi tipas nustatomas pagal tai, kuris parametras pasikeis pirmiausia [12].

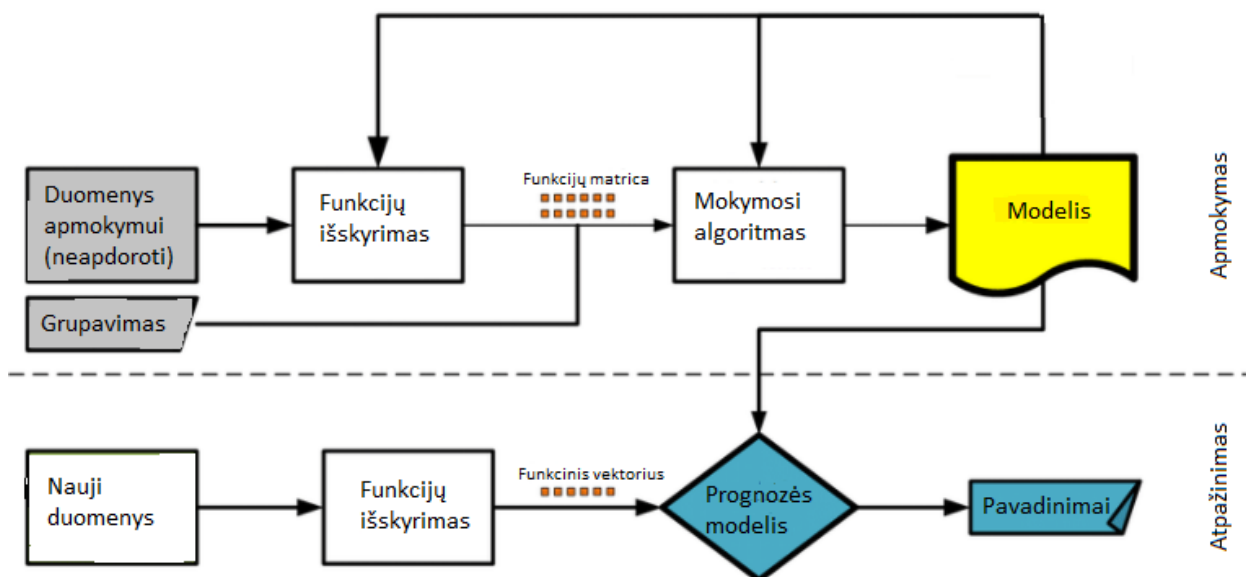
4.1.1. Prižiūrimas mokymasis

Prižiūrimas mokymasis (angl. *supervised learning*) arba asociatyvusis mokymasis vyksta tuomet, kai tinklas yra apmokomas, suteikiant jam įvesties ir išvesties atitikimo modelius. Šias įvesties ir išvesties poras gali suteikti išorinis mokytojas arba sistema, kurioje yra neuroninis tinklas [12]. Apmokomo tinklo modelio veikimas pavaizduotas 13 paveiksle.



13 pav. Apmokomo tinklo modelis [12]

Klaidų taisymo mokymosi algoritmas, remdamasis įvesties ir išvesties pavyzdžiais, treniruoja tinklą ir randa klaidos signalą, kuris yra apskaičiuotos išvesties ir norimo išėjimo skirtumas. Koreguojami neuronų sinapsiniai svoriai, proporcingi klaidos signalo sandaugai ir sinapsinio svorio įvesties egzemplioriui [13]. 14 paveiksle pavaizduota prižiūravimo mokymosi algoritmo klasifikavimo darbo eiga. Pirmajame etape neapdoroti mokymo duomenys išvalomi, iš visų galimų išrenkamos tik naudingos funkcijos. Tada žinomos klasės ir išgautos funkcijos perduodamos mokymo etapui, kur mašininio mokymosi algoritmai naudojami norint nustatyti nebrotuotą modelį, susiejantį įvestis su norimais rezultatais.

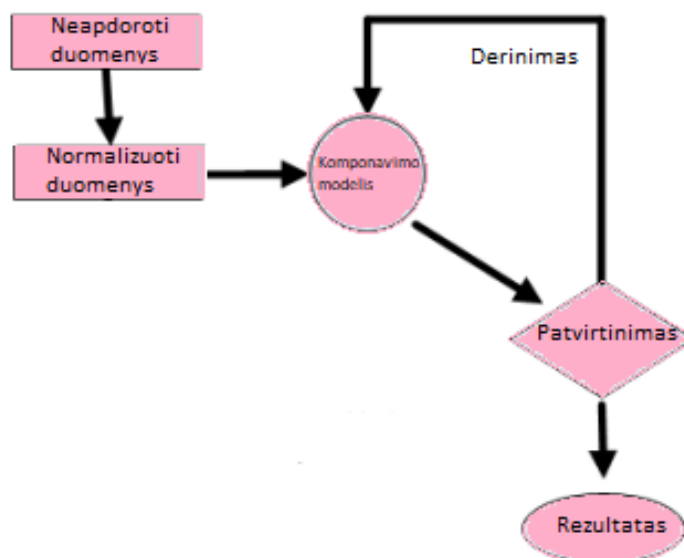


14 pav. Prižiūravimo tinklo mokymosi ir naudojimo schema [14]

Vertinimas suteikia grįžtamąjį ryšį apie funkcijų išgavimą ir mokymosi etapus, kad būtų galima koreguoti modelio tikslumą. Mokymo procesas kartojamas tol, kol pasiekiamas norimas tikslumo lygis. Suktas modelis naudojamas naujų duomenų etiketei numatyti [14].

4.2. Neprižiūrimas mokymasis

Neprižiūrimas mokymasis (angl. *unsupervised learning*) ar savęs organizavimas, kai išvesties vienetą yra mokomas reaguoti į įvesties modelio klasterius. Šioje paradigmoje sistema turi atrasti statistiškai svarbiausius įvesties populiacijos bruožus. Skirtingai nuo prižiūrimos mokymosi sistemos, savaiminis mokymasis neturi nustatyto kategorijų rinkinio, į kurį turėtų būti klasifikuojami modeliai. Sistema, priklausomai nuo įvesties domenų tipo, turi kurti savo klasifikavimo metodą [12]. Neprižiūrimo tinklo mokymosi schema pateikta 15 paveiksle.



15 pav. Neprižiūrimo tinklo mokymosi schema [15]

Neprižiūrimo mokymosi metu vyksta duomenų organizavimas į grupes, kai toje pačioje grupėje esantys duomenys yra panašūs vienas į kitą, o skirtingų grupių duomenys nėra panašūs.

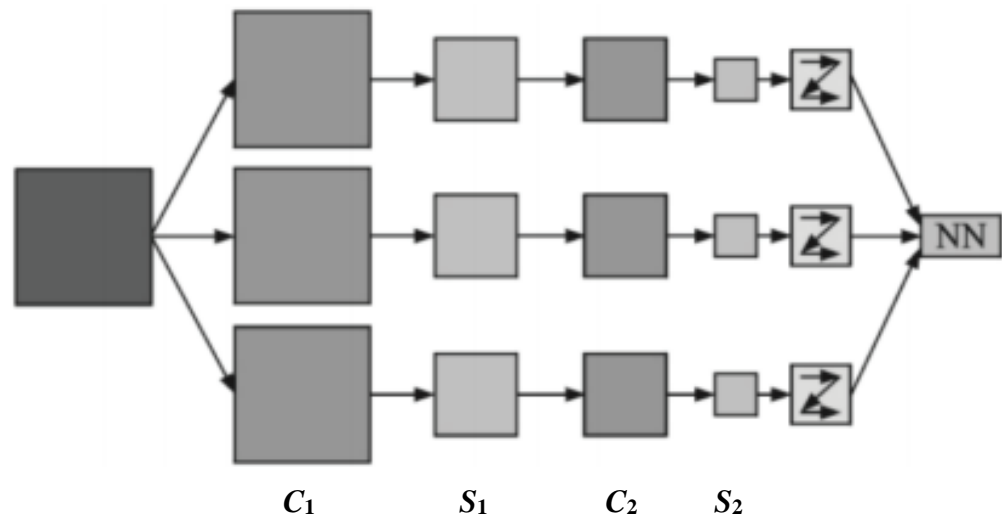
4.2.1. Sustiprintas mokymasis

Sustiprintas mokymasis (angl. *reinforced learning*) yra tokio tos tinklo apmokymo būdas, kuris gali būti laikomas tarpine aukščiau nurodytų dviejų mokymosi rūšių forma. Besimokanti mašina atlieka tam tikrus veiksmus aplinkoje ir iš jos gauna grįžtamąjį ryšį. Mokymosi sistema įvertina savo veiksmus: „gerai“ arba „blogai“, pagal reakciją į aplinką ir atitinkamai pakoreguoja savo parametrus. Paprastai parametru koregavimas tęsiasi tol, kol atsiranda pusiausvyros būseną, po kurios sistemos parametrai nebebus keičiami [12].

4.3. Konvoliucinis neuroninis tinklas

Vienas iš populiariausių neuroninių tinklų – konvoliucinis neuroninis tinklas (CNN). Šis pavadinimas paimtas iš matematinės tiesinės operacijos tarp matricių, vadinamos konvoliucija. CNN turi kelis sluoksnius – konvoliucinį, ne tiesiškąjį, kaupimo ir visiškai sujungtą [16]. Konvoliuciniai neuroniniai tinklai CNN yra analogiški tradiciniams ANN, nes juos sudaro neuronai, kurie patys optimizuojasi mokymuisi. Kiekvienas neuronas gauna įvestį ir atlieka operacijas (pvz., skaliarinė sandauga, po kurios eina netiesinė funkcija), kurios yra beribio dirbtinio neuroninio tinklo pagrindas. Nuo įvesties neapdorotų vaizdo vektorių iki galutinės klasės rezultato išvesties, tinklas išreikš vieną rezultato

funkciją (svorį) [17]. Paprasta konvoliucinio neuroninio tinklo modelio struktūros schema parodyta 16 paveiksle. Tinklo modelis susideda iš dviejų konvoliucinių sluoksnių (C_1 , C_2) ir dviejų dalinai pavyzdinių sluoksnių (S_1 , S_2). Pirma, pradinį įvesties vaizdą sujaukia trys apmokyti filtrai (vadinami konvoliuciniu branduoliu) ir pridėtiniai vektoriai. C_1 sluoksnyje sugeneruojami trys funkcijų žemėlapiai, o po to kiekvienam objektų žemėlapiui nustatomi vietiniai regionai ir jų vidurkis [18].



16 pav. Konvoliucinis neuroninis tinklas [18]

Naudojant netiesinę aktyvavimo funkciją S_1 , sluoksnyje gaunami trys nauji žemėlapiai. Šie funkcijų žemėlapiai yra sujungti su trimis apmokytais C_2 sluoksniu filtrais, o trys funkcijų žemėlapiai išvedami per S_2 sluoksnį. Galutinis S_2 sluoksniu išėjimas yra vektorizuojamas ir tada įvedamas į tradicinį neuroninį tinklą, mokymuisi [18].

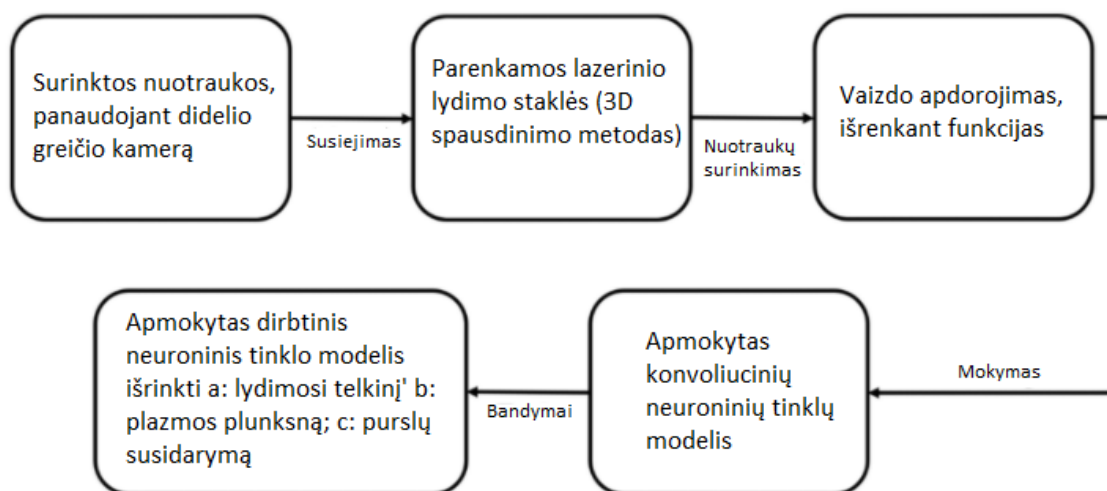
5. Dirbtinio neuroninio tinklo panaudojimas 3D spausdinime

3D spausdinimo procesas sujungia kelis aspektus: modelio dizainą, medžiagos pasirinkimą, spausdinimo procesą, pagamintų dalių vertinimą ir apibūdinimą. ANN taikomas 3D spausdinimo procese, atliekant procedūrų stebėjimą, projektavimą ir koreliacijos tarp proceso parametrų bei gauto komponento galutinių savybių nustatymą [19].

5.1. Proceso stebėjimas

Spausdinimo proceso stebėjimas naudojant įvairius jutiklius leidžia užtikrinti tiesioginės informacijos apie vykstantį procesą ir kokybės kontrolę rinkimą. Galima išskirti tris duomenų šaltinių tipus: a) vienmatis, pavyzdžiui, spektrai; b) dvimatis, pavyzdžiui, grafikai ir vaizdai; ir c) trimatės, struktūros ar formos. Vienmačius duomenis galima apdoroti greičiau, tačiau jie teikia mažiau informacijos, palyginti su dviejų ir trijų matmenų duomenimis [19].

Zhang'as ir kt. [19] aprašė ir ištyrė metodą, skirtą proceso vaizdams vaizdo kamera rinkti. Tokia sistema gali būti vadinama vaizdine sistema. Ši sąranka renka informaciją apie: a) lydinio telkinį, b) plazmos plunksną ir c) pūslų susidarymą. Funkcijos buvo paimtos iš vaizdų, jas įvedus į įprastą ML algoritmą. Autorių pasiūlytas algoritmas pateiktas 17 paveiksle. Be to, autoriai pastebėjo, kad nors konvoliuciniam ANN modeliui nereikia funkcijų išskyrimo, jis vis tiek gali numatyti rezultatus 92,7 procentų tikslumu. Tai paverčia konvoliucinį ANN potencialiu kandidatu 3D spausdinimo procesui realiuoju laiku stebėti [19].



17 pav. Konvoliucinio neuroninio tinklo algoritmas [19]

Norint nustatyti trūkumus naudojant žmogaus sukurtus ir sąlygomis pagrįstus algoritmus, reikia išsamiai išnagrinėti spausdinimo procesą bei turėti žinių apie mašininį vaizdavimą. Tokiomis sąlygomis pagrįsti algoritmai yra labiau ribojantys; nauji algoritmai turi būti sugeneruoti, kai atsiranda naujų medžiagų arba kai įvedama naujų dalių geometrija, nes taikant šį metodą reikia atsižvelgti į įvairių parametrų sąveiką. Pasikliaudamas žmogaus sukurtomis sąlygomis, pagrįsti algoritmai tampa ne tokie praktiški [19]. ANN leidžia defektus aptikti naudojant didelį gerų ir blogų spausdinimo pavyzdžių duomenų rinkinį, o aptikimo galimybes galima pagerinti pridėdant naujų mokymo duomenų.

6. Kameros kalibravimas

Plataus kampo, daugiakryptės ir „žuvies akies“ kameros šiais laikais yra populiaros daugelyje kompiuterinio matymo, robotikos ir fotogrametrinių užduočių, tokių kaip navigacija, lokalizavimas, sekimas, žemėlapiai ir pan. sistemose. Vaizdo kameros kalibravimas yra būtinas, norint išgauti metrinę informaciją iš nuotraukos ir nustatyti santykį tarp 3D spindulio bei jo vaizdo plokštumoje [20].

6.1. Kalibravimo metodai

Kameros kalibravimas susideda iš dviejų žingsnių. Pirmasis apima tinkamo modelio: dioptrinio („žuvies akies“) ar katadioptrinio (įvairiakrypčio), apibūdinančio vaizdo sistemos elgseną, pasirinkimą. Antrasis žingsnis, visų parametrų (sistemos vidinė ir išorinė orientacija, iškraipymo parametrai) įvertinimas. Siūlomi keli kalibravimo metodai, kuriuos sąlyginai galima suskirstyti į tris kategorijas, pradedant nuo paprasčiausio [20].

6.1.1. Automatinis kalibravimas

Automatinis arba savaiminis kalibravimas, kai naudojami epipoliniai apribojimai ir taškų atitikimai tarp kelių vaizdų. Autokalibravimas dioptrinėms ir katadioptrinėms sistemoms yra pagrįstas epipolinės geometrijos nustatymu iš kelių atitikmenų dviejuose vaizduose. Šiam modeliui, RANSAC (žr. 18 pav.) naudojamas pašalinti neįprastus rodiklius, apie vaizdą nereikalingi jokie išankstiniai duomenys (objekto matmenys, ar kiti kalibravimo metu reikalingi duomenys nustatyti plokštumos plotą). Vėliau, norint nustatyti visus kameros parametrus ir rekonstruoti vaizdą, gali būti naudojamas paketo bloko koregavimas [21].

Duomenų rinkinio pavyzdys, sudarytas iš teisingų duomenų ir nedidelio kiekio nenormalių duomenų. Taikant RANSAC algoritmą funkcijų taškų atitikimo atrankai, galima efektyviai pašalinti klaidų atitikimo taškus.



18 pav. RANSAC algoritmo vizualizacija [21]

Tačiau RANSAC algoritmas taip pat turi trūkumų, kuriuos sukelia visų suderinamų taškų porų pasikartojamumas [21].

6.1.2. Neplaninio objekto kalibravimas

Neplaninis objekto kalibravimas – tai vieno vaizdo su žinomomis 3D pasaulio koordinatėmis kalibravimas, tiriant „žuvies akies“ lęšių geometrines projekcijas. Kamera kalibruojama, naudojant vieną kambario nuotrauką, paruoštą naudojant 3D taškus, kurie apima „žuvies akies“ objektyvo matymo lauką (žr. 19 pav.).



a)

b)

19 pav. „Žuvies akies“ efektas [22]: a) prieš kalibravimą; b) po kalibravimo

Kairėje paveikslo pusėje (žr. 19 pav.) matomas „žuvies akies“ efektas prieš kalibravimą, o dešinėje pusėje (žr. 19 pav.) to paties kambario nuotrauka po atlikto kalibravimo. Norint pagerinti modelio kokybę ir kompensuoti realaus vaizdo nukrypimus nuo geometrinio modelio, į projekcijos modelį įtraukiami radialiniai ir tangentiniai iškreipimų parametrai. Siūlomas katadioptrinių fotoaparatu kalibravimo metodas, naudojant erdvės kalibravimą. Pakeltos koordinatės yra naudojamos linijiškai nukreipti vaizdo taškus į nuotraukos plokštumą. Įvertinus projekcijos matricą iš vaizdo taškų, paskirstytų per 3D kalibravimo objektą, taikomas galutinis netiesinis patikslinimas [22].

6.1.3. Kalibravimas naudojant šaškių lentą

Trečiasis kalibravimo metodas apima plokščio objekto (pvz., šaškių lentos) su žinomomis 3D objekto koordinatėmis, stebėjimą. Šiam kalibravimui atlikti reikalingos bent dviejų plokštumos objektų nuotraukos. „Žuvies akies“ lęšiams naudojamas nuotraukos centras. Apibendrintas židinio nuotolis, kuris yra tiksli projekcijos modelio dalis, įvertinamas bent iš trijų tiesės taškų. Norint automatiškai išgauti visus šaškių lentos kampus, planinė homografija įvertinama iš keturių rankiniu būdu pasirinktų taškų. Tada likusieji šaškių lentos taškai suprojektuojami į numatomą jų vietą ir atliekama vietinė pikselių kampo paieška. Iš pradžių įvertinta homografija yra pagrįsta pradinėmis reikšmėmis, tokiomis kaip apibendrintas židinio nuotolis ir iškreipimo centras. Kalibravimo procedūra, pagrįsta keliais plokščio kalibravimo objekto stebėjimais, yra visiškai automatinė ir nereikalauja tolesnio vartotojo sąveikos ar išankstinių žinių. Galimas trūkumas yra neteisingas taškų ištraukimas, kuris gali turėti įtakos galutiniam kalibravimo rezultatui [20]. Taip gali nutikti, kai panaudojama kalibravimo taško koordinatė nesutampa su naudojama nuotraukoje.

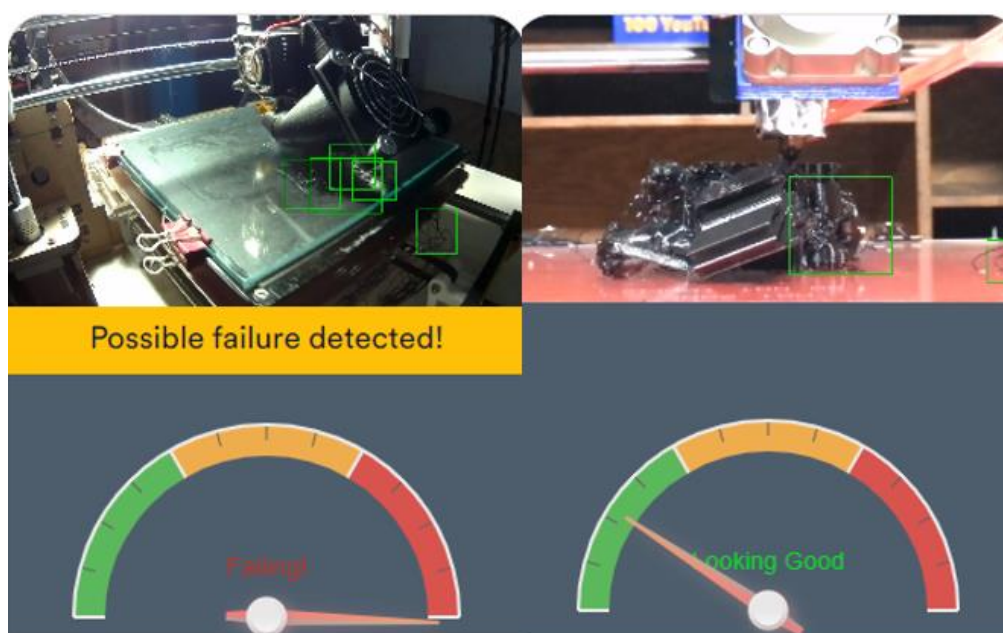
7. Eksperimentinė dalis

7.1. Kuriamos ir rinkoje esančios sistemų palyginimas

Šiuo metu rinkoje egzistuoja keletas sistemų, kurios daugiau žinomos – „Printpal“ ir „The Spaghetti Detective“, kurios taikydamos dirbtinį intelektą aptinka 3D spausdinimo broką. Tokio tipo sistemos aptinka tik 1 rūšies broką ir pasižymi savybe stabdyti spausdinimą reikiamu metu.

Šiame projekte kuriamos broko atpažinimo sistemos tikslas – gebėti identifikuoti kelias skirtingas defektų rūšis, pakoreguoti parametrus realiu laiku, tęsiant darbą, o stabdymą atlikti tik įvertinus, kad parametru koregavimas nepadės ištaisyti esančios(-ių) klaidos(-ų).

Palyginimui pasirinktas „OctoPrint“ programavimo įrankis „The Spaghetti Detective“. Šis įrankis atpažįsta ištemptos siūlės defektą (žr. 20 pav.). Kiekvienas aptiktas defektas apvedamas, stačiakampe figūra. Programoje naudojama 3 dalių sekcija su rodykle. Spalvota indikacija parodo spausdinimo būseną. Jei rodyklė ties žalia spalva, reiškia, kad spausdinimas vyksta teisingai. Jei geltonoje zonoje, kad galimai yra aptiktas defektas, bet spausdinimas tęsiamas. Jei pasiekta raudona zona, spausdinimas stabdomas ir mobiliojo įrenginio programėlėje išsiunčiama žinutė su nuotrauka bei užklausa ar tęsti spausdinimą ar galutinai sustabdyti.



20 pav. Aptikti spausdinimo defektai [23] (nustatytas galimas gedimas – angl. „possible failure detected“; klaida – angl. „failing“; tinkamas – angl. „looking good“)

Šioje sistemoje nustačius broką, yra numatytas arba tiesioginis spausdinimo stabdymas, arba tekstinio pranešimo su pasirinkimu tęsti arba stabdyti procesą ekrane pasirodymas.

7.2. Fizinė įranga

Broko atpažinimo sistema projektuojama „Ender 3Pro“ 3D spausdintuvo bazėje. Spausdintuvo parametrai: 220 × 220 × 250 mm kaitinamas, magnetinis pagrindas, kurio maksimali temperatūra 110 °C; kaitinimas ekstruderis, kurio rekomenduojama maksimali temperatūra 260 °C. Spausdinimo

greitis iki ≤ 180 mm/s, normalus spausdinimo greitis 30–60 mm/s, pakartojamumas $\pm 0,1$ mm. Naudojamo plastiko skersmuo 1,75 mm. Maitinimo šaltinis 24 VDC, 360 W.


Naudojama fizinė įranga:

- mikrokompiuteris „Raspberry Pi“ 4b;
- kamera;
- šviesos šaltinis;
- įrangos laikikliai.

7.2.1. Mikrokompiuteris „Raspberry Pi“ 4b

Mikrokompiuteris (angl. *single-board computer*, SBC) naudojamas komandoms į 3D spausdintuvą siųsti ir iš jo nuskaityti, taip pat vaizdinei medžiagai apdoroti, apšvietimui kontroliuoti bei neuroniniam tinklui apmokyti. Naudojamo mikrokompiuterio specifikacija nurodyta 1 lentelėje.

1 lentelė. „Raspberry Pi“ 4b specifikacija [24]

Keturių branduolių 64-bitų „ARM-Cortex“ A72, mikroprocesorius, veikimo dažnis 1,5 GHz	
4 Gigabitų LPDDR4 RAM	
H.265 didelio efektyvumo vaizdo kodavimas (iki 4Kp60)	
H.264 išplėstinis vaizdo įrašų kodavimas (iki 1080p60)	
„VideoCore“ VI 3D grafikos	
Palaiko 2 HDMI ekranus iki 4Kp60	
28 GPIO kontaktai, skirti įvairioms sąsajoms, kaip: UART, I2C, SPI	

7.2.2. Kameros parinkimas

Kamera yra skirta vaizdo medžiagai išgauti, kai fotografuojamas ar filmuojamas objektas, esantis uždaroje patalpoje su tamsia fono medžiaga, nejuda. Fotografuojant turi būti naudojamas šviesos šaltinis. 2 lentelėje pateiktas dviejų skirtingų kamerų palyginimas.

2 lentelė. „Raspberry Noir“ V2 ir „Logitech“ C920E palyginimas [25, 26]

Kameros modelis	„Raspberry Noir“ V2	„Logitech“ C920E
Raiška (MP)	8	3
Svoris (g)	3	162
Rezoliucija (pikseliais)	3280 × 2464	1920 × 1080

Vaizdo kodavimas	H.265/H.264	H.264
Vaizdo įrašo rezoliucija	1080p30	1080p30
Matmenys	25 × 23 × 9 mm	94 × 43,2 × 71,1 mm
Jungties tipas	SFW15R-2STE1LF	USB-A
Kaina	25 €	70 €

„Raspberry Noir“ V2 kamera pasižymi daugiau nei 2,5 karto didesne raiška lyginant su „Logitech“ C920E kamera, o specifinis jungties tipas leidžia sutaupyti USB jungčių. Be to šio modelio kameros modulis yra kelis kartus pigesnis. Pagrindinis *Logitech* kameros privalumas – uždara ir apsaugota nuo išorinių veiksnių. Pasirinkta kamera – „Raspberry Noir“ V2, nes ji pigesnė, fiziškai mažesnė ir yra didesnės raiškos.

7.2.3. Šviesos šaltinis

Kadangi tikrinamas objektas yra tamsioje patalpoje, norint gauti ryškų jo vaizdą kadre, reikalingas šviesos šaltinis, galintis apšviesti detalę iš kelių skirtingų kampų, siekiant sumažinti šešėlius iš vaizdo kameros matymo lauko. Projektuojamai sistemai reikalingas 0,048 m² apšviečiamas plotas.

Nustatant koks apšvietimo intensyvumas reikalingas, buvo matuotas liuksų kiekis spausdinamame paviršiuje. Bandymo metu pakabinamas šviesos šaltinis, reguliuojamas atstumas iki fotografuojamo objekto ir palyginama, kuriuo būdu geriausiai matoma spausdinimo kokybė. Bandymai atlikti naudojant 24V DC, 100 mm ilgio LED juostą, nuotraukų rezoliucija 1920 × 1080 taškų.

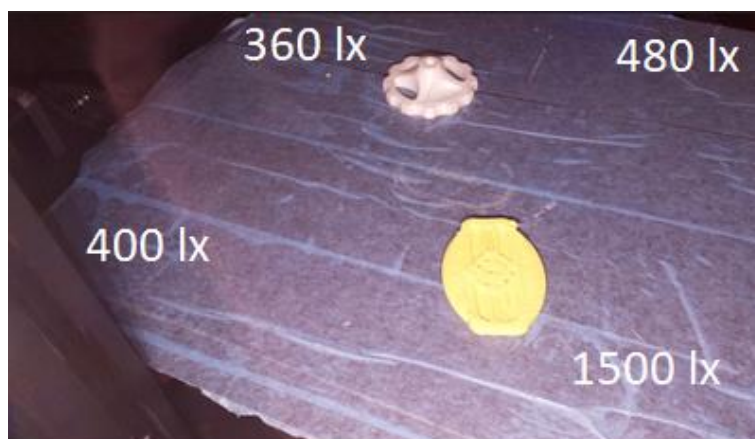
1 bandymas: kai šviesos šaltinis tvirtinamas 380 mm aukštyje, statmenai fotografuojamam paviršiui, o kamera įtaisyta ant Z ašies variklio gaubto, 100 mm atstumu nuo fotografuojamo paviršiaus centro. Pasinaudojant „Lux Light Meter“ programėle išmaniajame telefone pamatuotas šviesos srautas tenkantis plokštumai keturiuose jos kampuose. 21 paveiksle pavaizduotas bandymo rezultatas.



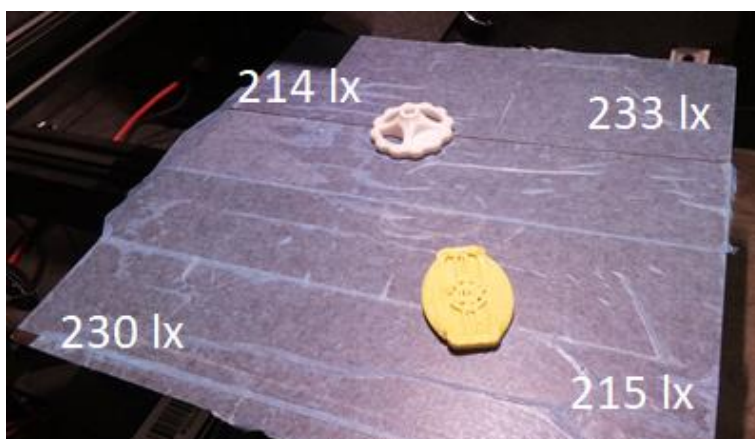
21 pav. 1 bandymo apšviestas plotas

2 bandymas: kai šviesos šaltinis ir kamera tvirtinami ant Z ašies variklio gaubto, 100 mm atstumu nuo fotografuojamo paviršiaus centro. Pamatuotas šviesos srautas, tenkantis plokštumai, keturiuose kampuose. 22 paveiksle pavaizduotas bandymo rezultatas.

3 bandymas: kai šviesos šaltinis tvirtinamas 600 mm aukštyje, statmenai fotografuojamam paviršiui, o kamera montuojama ant sienos, 220 mm atstumu nuo fotografuojamo paviršiaus centro. Pamatotas šviesos srautas tenkantis plokštumai, keturiuose kampuose. 23 paveiksle pavaizduotas bandymo rezultatas.



22 pav. 2 bandymo apšviestas plotas



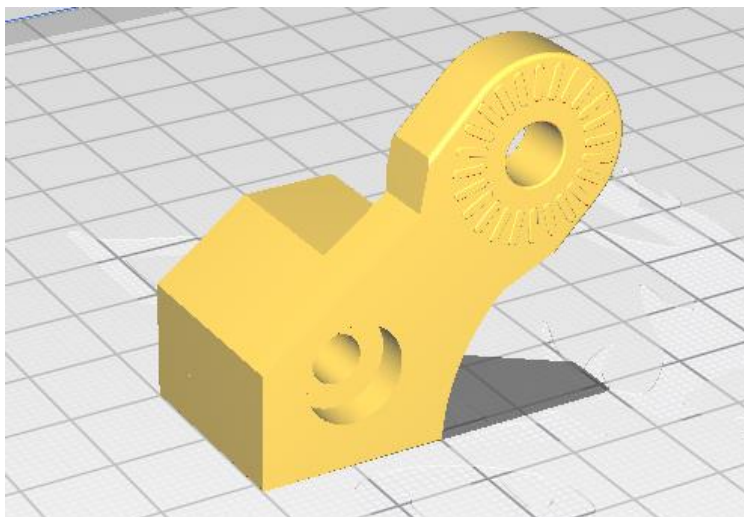
23 pav. 3 bandymo apšviestas plotas

Atliekant bandymus pastebėta, kad esant vienodam paviršiaus apšviestumui, fotografuojamo objekto kontūrai matomi ryškiau. Iš paveikslo (žr. 22 pav.) matyti, kad objektai, esantys arčiau šviesos šaltinio, apšviečiami ryškiau ir netolygiai, todėl detalė atsispindi ir jos kontūrai tampa mažiau matomi.

7.2.4. Įrangos laikikliai

Parinkta kamera nėra pritaikyta tiesiogiai montuoti ant 3D spausdintuvo rėmo. Jos plokštės komponentai nėra izoliuoti nuo dulkių ar kitų nenumatytų veiksnių. Panaudotas kameros laikiklio dizainas [27]. Alkūnės sujungtos $M6 \times 0,75$ žingsnio plastikiniais varžtais ir $M6$ veržlėmis, kurios susijungia į šešiakampę alkūnės skylę ir užsifiksuoja. Kameros korpusas sudarytas iš 2 sujungiančiųjų dalių su skylė skirta laidui. Korpuso jungiamoji dalis su alkūne tvirtinama naudojant rutulio formos rankeną, kuri įsitvirtina į šešiakampę įdubą. Tvirtinimo sujungimas su spausdintuvo rėmu (žr. 24 pav.) modifikuotas taip, kad galima būtų panaudojant varžtą priveržti laikiklį prie x ašies rėmo.

25 paveiksle pateiktas atspausdintas kameros ir apšvietimo laikiklis. Kadangi apšvietimas buvo pasirinktas naudojant LED juostą, atspausdintas 100 mm ilgio, 135° kampu išlenktas laikiklis.



24 pav. Modifikuotas laikiklis



25 pav. Kameros ir apšvietimo laikiklis

Laikiklio alkūnės ir kameros korpuso dalys spausdintos su 30 proc. užpildu. Plastikiniams veržlėms ir varžtams bei jų sriegiams naudotas 100 proc. užpildas.

7.3. Programinė įranga

Didžioji broko atpažinimo sistemos programinės įrangos dalis yra naudojama vaizdinei medžiagai apdoroti, komunikacijai su spausdintuvu palaikyti ir neuroniniam tinklui naudoti bei apmokyti. Darbui su grafinio dizaino programomis naudojamas nešiojamas kompiuteris „Asus“ X555L.

Naudojama programinė įranga:

- „Raspbian Debian“ 10, 32-bitų operacinė sistema;
- „TensorFlow“ 2.0.0 programa;
- „Ultimaker Cura“ 4.6 programa;

- „Autodesk Fusion“ 360 programa.

7.3.1. 32-bitų operacinė sistema „Raspbian Debian“ 10

„Raspberry Pi OS“ – tai nemokama operacinė sistema (pagrįsta „Debian“), kuri optimizuota „Raspberry Pi“ techninei įrangai. Sistemoje yra daugiau nei 35 000 paketų [28].

„Raspbian Buster“, išleista kaip „Raspbian Stretch“, buvo numatytoji „Raspberry Pi“ operacinė sistema beveik 2 metus ir yra suderinama su visais „Raspberry Pi“ modeliais. „Buster“ veikia „Debian Linux“ 10 versijos pagrindu, kur naudojama atvirojo kodo 4.19 „Linux“ versija ir 8.3 GCC kompiliatorius [28]

Naudojamos bibliotekos:

- „Numpy“ – matematinė biblioteka matricoms skaičiuoti;
- „GPIO“ – įgalinamas fizinių signalų naudojimas;
- „Keras“ – giliojo mokymosi aplikacija, palengvinanti „TensorFlow“ naudojimo galimybes;
- „PySerial“ – leidžia pasinaudoti įvairiais prietaisų komunikacijos būdais.

7.3.2. „TensorFlow“ 2.0.0 biblioteka

„TensorFlow“ yra programinės įrangos biblioteka, sukurta „Google Brain“ komandos, priklausančios „Google“ mašininio mokymosi tyrimų organizacijai. „TensorFlow“ sujungia kompiliavimo optimizavimo metodų skaičiavimo algebrą, todėl atsiranda galimybė apskaičiuoti įvairias matematinės išraiškas, kai problema yra laikas, reikalingas skaičiavimui atlikti. Pagrindinės savybės: matematinė išraiškų, apimančių daugiamatį masyvą (tenzorijų), apibrėžimas, optimizavimas ir efektyvus apskaičiavimas. Skaičiavimams naudojant GPU, automatizuojamas tos pačios atminties ir naudojamų duomenų valdymas bei optimizavimas [29]. Yra galimybė tą patį kodą paleisti naudojant CPU arba GPU. „TensorFlow“ išsiaiškina, kurios skaičiavimo dalys turėtų būti perkeltos į GPU.

7.3.3. „Ultimaker Cura“ 4.6 programa

„Ultimaker Cura“ programa „pjausto“ vartotojo modelio failą į sluoksnius ir generuoja spausdintuvo specifinį G-kodą, kuris gali būti siunčiamas į spausdintuvą fizinio objekto gamybai. Tai atviro kodo programinė įranga suderinama su dauguma 3D spausdintuvų, gali dirbti su tokiais 3D formatais: STL, OBJ, X3D, 3MF, tokiais vaizdo failų formatais: BMP, GIF, JPG ir PNG.

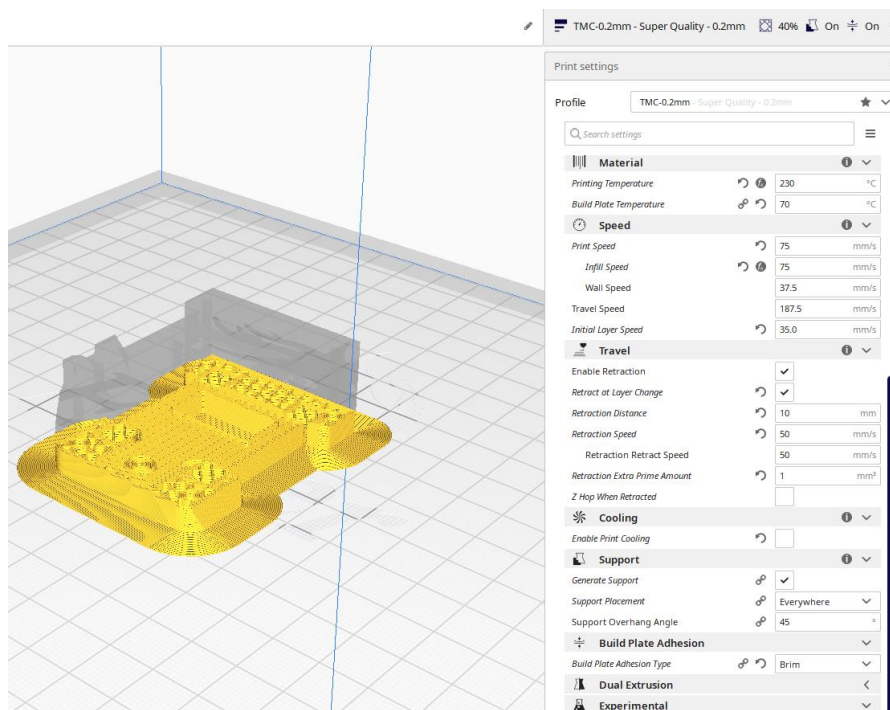
26 paveiksle pateikta „Ultimaker Cura“ aplinka, kurioje apdorojamas 3D modelis ir konvertuojamas į G-kodą.

Vartotojai gali sukurti savo papildinius arba naudoti komerciškai prieinamus. Papildiniai supaprastina darbo eigą vartotojams, leidžiant atlikti tokias užduotis, kaip atidaryti failą iš meniu arba eksportuoti failą iš paraiškos.

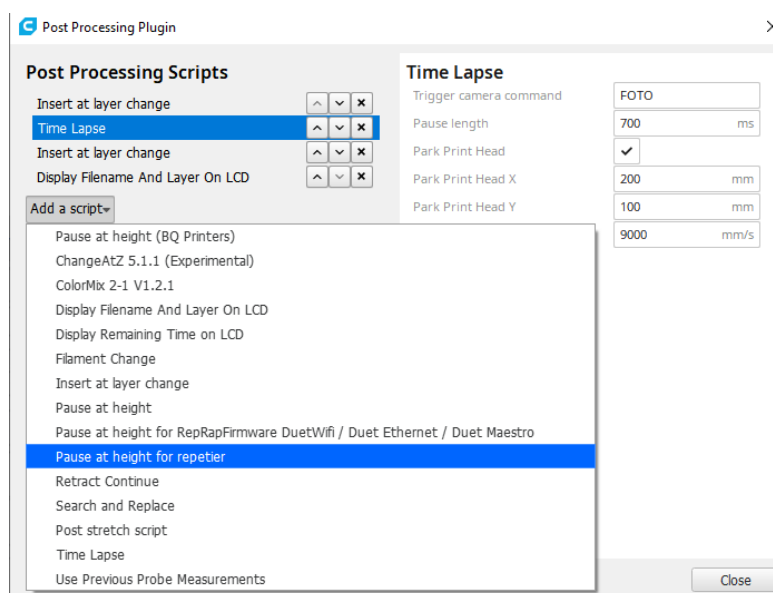
27 paveiksle pavaizduoti „Ultimaker Cura“ papildiniai, kuriais naudojantis sugeneruojamas individualius poreikius atitinkantis G-kodas.

Pagrindiniai parametrai matomi 26 paveiksle nustato kaitinamo ekstruderio ir pagrindo temperatūras, judėjimo greičius mm/s spausdinant išorines, vidines gaminio dalis bei judant į koordinates, kai nėra tiekiamas plastikas.

Pasirenkama ar reikalingas aušinimas, ar spausdinant reikia sugeneruoti papildomas konstrukcijas, skirtas sustiprinti gaminį, jo spausdinimo metu.



26 pav. Modelio peržiūra



27 pav. „Ultimaker Cura“ papildiniai

Naudojant 27 paveiksle matomus papildinius („Time Lapse“, „Insert at Layer Change“), generuojamas G-kodas, kuris po kiekvieno sluoksnio judina X ašį 200 mm atstumu, o Y ašį – 100 mm atstumu. Po šių judesių seka 700 ms pauzė ir daroma kiekvieno sluoksnio nuotrauka.

7.4. Nuotraukų kaupimas

Nuotraukoms kaupti naudojamas programinis kodas (žr. 27 pav.), kuriuo skaitomas atsakymas iš 3D spausdintuvo, kada galima fotografuoti. Skaičiuoklėje pavadinimu „file“ kaupiamas padarytų

nuotraukų skaičius. Kiekvieno fotografavimo metu perrašoma paskutinė reikšmė. Spausdintuvo komunikacijoje negalima matyti kiekvienos komandinės eilutės, todėl pasinaudojant „Cura“ papildiniu sugeneruojamas „String“ formato atsakymas, kai yra galimas fotografavimas. Nuskaitytus leidimą fotografuoti, įjungiamas apšvietimas panaudojant „RaspberryPi GPIO“ kontaktus, apverčiamas kameros vaizdas, panaudojamas pilkos spalvos efektas ir nuotrauka išsaugoma su priskirtu numeriu.

7.4.1. Žodinis algoritmas nuotraukoms kaupti

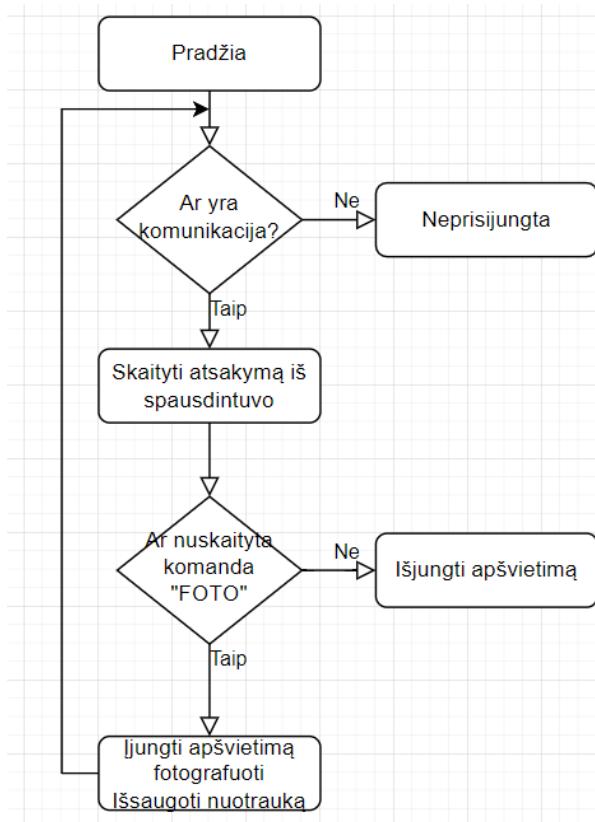
Pirmame žingsnyje tikrinama USB komunikacija, tuomet:

- jei neprisijungta, terminalo lange parašomas įspėjimas.
- Jei prisijungta, nuskaitytas gautas atsakymas ir laukiama specifinio „String“ formato komandos „FOTO“.

Toliau, jei komanda gauta: įjungiamas apšvietimas, nuskaitytas failas, kuriame kaupiamas nuotraukų skaičius; prie esamo nuotraukų skaičiaus pridedamas vienetas ir išsaugomas failas. Paskutiniame žingsnyje fotografuojama, nuotraukai priskiriamas toks pats numeris, kaip išsaugotame faile. Jei gauta komanda nesutampa su laukiama – išjungiamas apšvietimas.

7.4.2. Algoritmas nuotraukoms kaupti

Struktūrinis nuotraukų kaupimo algoritmas pavaizduotas 28 paveiksle, o programinis nuotraukų kaupimo kodas – 29 paveiksle.



28 pav. Nuotraukų kaupimo algoritmas

```

SerialEnder3.py **
1  import serial
2  import time
3  import RPi.GPIO as GPIO
4  import time
5  import picamera
6  GPIO.setmode(GPIO.BCM)
7  GPIO.setwarnings(False)
8  GPIO.setup(4,GPIO.OUT)
9
10 file = '/home/pi/tf/counter.txt'
11 if __name__ == '__main__':
12     ser = serial.Serial('/dev/ttyUSB0', 115200, timeout=None)
13     ser.flush()
14     while True:
15         line = ser.readline().decode('utf-8-sig').rstrip()
16         #line = ser.readline().decode('utf-8')
17         print (line)
18         if line ==('echo:Unknown command: "FOTO "'):
19             GPIO.output(4,GPIO.HIGH)
20             with open(file, 'r') as f:
21                 line = f.read()
22                 num = int((line.split())[0])+1
23                 #overwrite number (counter)
24             with open(file, 'w') as f:
25                 f.write(str(num))
26             with picamera.PiCamera() as camera:
27                 camera.rotation = 180
28                 #camera.color_effects = (128,128)
29                 camera.capture("/home/pi/tf/image" + str (num) + ".jpg")
30         else:
31             GPIO.output(4,GPIO.LOW)

```

29 pav. Nuotraukų kaupimo programinis kodas

29 paveiksle pateiktas nuotraukų kaupimo kodas, kuriuo pasinaudojant kaupiamos nuotraukos skirtos neuroniniam tinklui mokyti.

7.4.3. Nuotraukos neuroninio tinklo mokymui

Aprašytas programinis kodas (žr. 29 pav.) leidžia gautas originalias spalvotas nuotraukas paversti į pilkų atspalvių nuotraukas. Sekančiame žingsnyje yra nurodoma vieta, kurioje matomas konkretus defektas ir tokia apdorota nuotrauka yra išsaugojama. Spausdinimo metu kaupiamos nuotraukos, kuriose matomi defektai: ištemptos siūlės, per didelis liejamo plastiko kiekis.

Spausdintų objektų nuotraukos, kuriose yra identifikuoti defektai, grupuojami į atskirus aplankus. Toliau, pritaikius 30 paveiksle pateiktą programinį kodą, nuotraukos su matomais defektais ir defekto koordinatėmis yra apkerpamos. Jei defektai matomi keliose tos pačios nuotraukos vietose, koordinatės nustatomos 2 kartus, o nuotrauka išsaugoma keliais etapais.

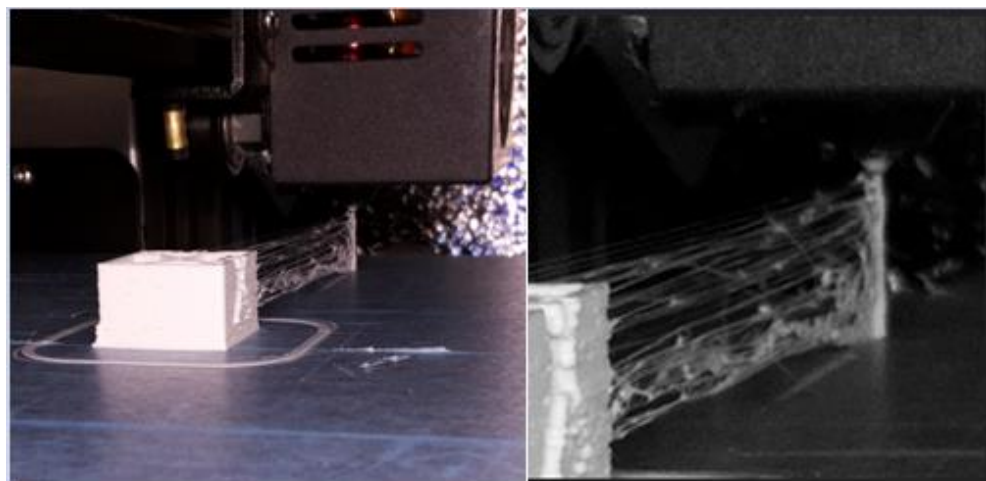
31 paveiksle ir 32 paveiksle pateiktos nuotraukos prieš ir po paruošimo neuroninio tinklo mokymui palyginimas. 31 paveikslo dalyse *a*, *c* ir *e* bei 32 paveikslo dalyse *a* ir *c* pavaizduotos originalios spalvotos nuotraukos, kurių raiškos 1920×1080 pikselių. Dalyse *b*, *d* ir *f* (žr. 31 pav.) bei dalyse *b* ir *d* (žr. 32 pav.) parodytos sumažintos nuotraukos su priartintais defektais, kurių raiškos 200×200 pikselių. Pateiktame 31 paveiksle pavaizduoti siūlės tempimosi defektai, o 32 paveiksle per didelio medžiagos kiekio defektai.

Neuroninio tinklo mokymui skirtos nuotraukos po pirminio apdorojimo grupuojamos į aplanką po 100 vnt. Skirtingose aplankuose sudėtos nuotraukos su skirtinga broko rūšimi. Vaizdai su defektais

rūšiuojami į 5 grupes: ištempta siūlė, išliejamas per didelį plastiką, užsikišęs kaitinamas ekstruderis, neprilipęs spausdinys prie pagrindo, sutrūkinėjęs paviršius.

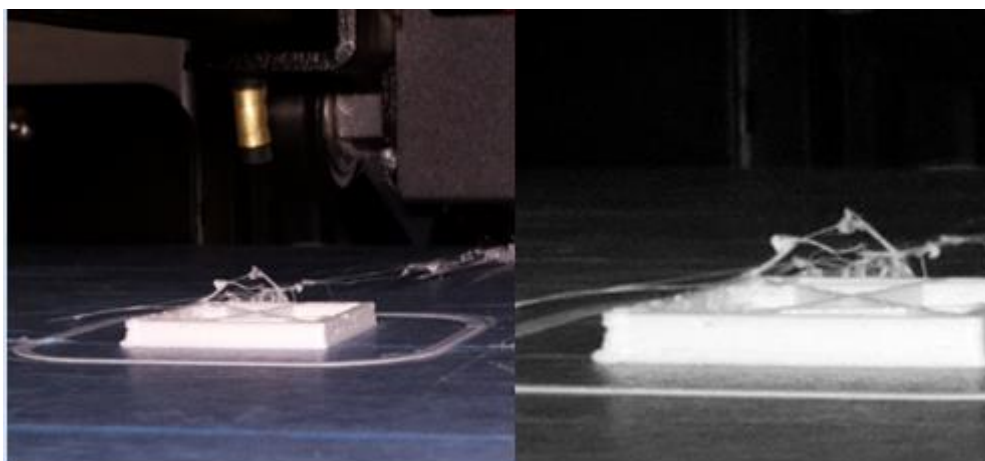
```
loopimageREAD.py ✖
1 import cv2
2 import glob
3
4 i=0
5 for img in glob.glob('/home/pi/NotCropped_im/OverExtrude' + '/*.jpg'):
6     image = cv2.imread(img, 0)
7     y=1050
8     x=300
9     h=1250
10    w=500
11    crop_image = image[x:w, y:h]
12    cv2.imwrite("/home/pi/Cropped_Images/OverExtrude/image%i.jpg" %i, crop_image)
13    i+=1
14    cv2.waitKey(30)
15    cv2.destroyAllWindows()
16
```

30 pav. Nuotraukų apkarpymas



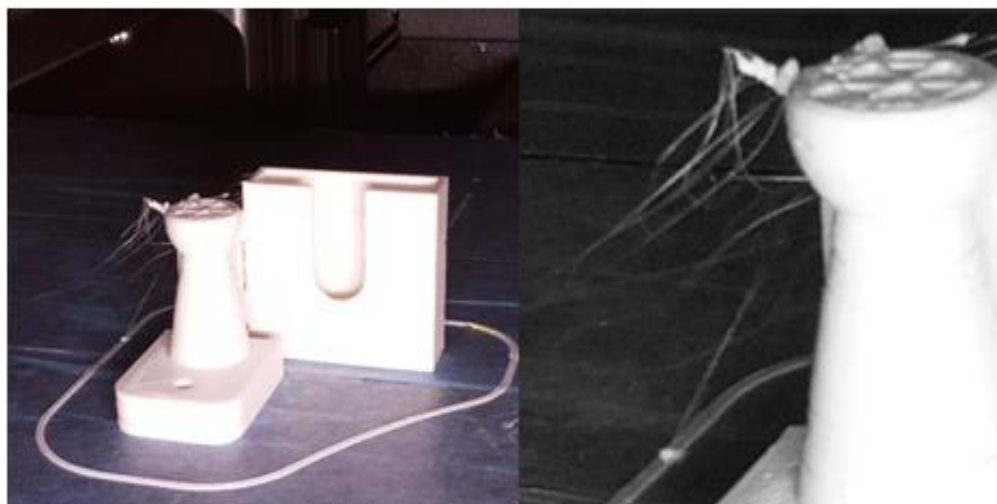
a)

b)



c)

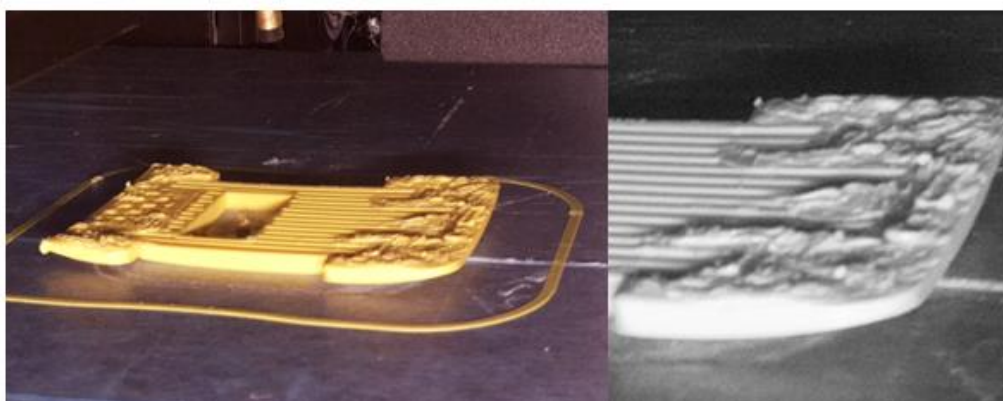
d)



e)

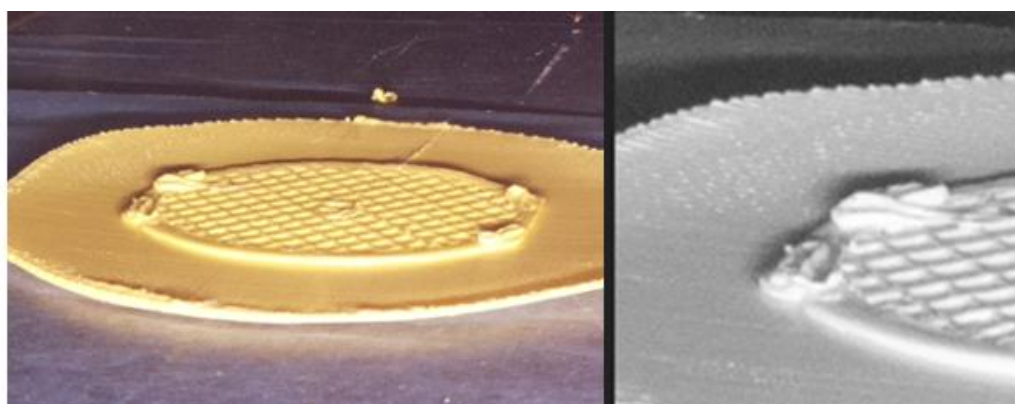
f)

31 pav. Palyginamos nuotraukos prieš apdorojimą (a, c, e) ir po apdorojimo (b, d, f)



a)

b)



c)

d)

32 pav. Palyginamos nuotraukos prieš apdorojimą (a, d) ir po apdorojimo (b, e)

Kiekviename aplankale sukaupta apie 100 nuotraukų, iš kurių apdorojimui panaudotos 70. Kadangi apmokymui naudojamas grupavimas aplankuose, nuotraukos turinčios kelias rūšis defektų grupuojamos pagal tai, kuris defektas yra geriau atpažįstamas.

7.5. Neuroninio tinklo apmokymas

Apmokomas tinklas nustatyti skirtingų rūšių 3D spausdinamų objektų defektus. Tinklui apmokyti naudojamas greito mokymosi ir optimizavimo algoritmas „Adam“. Nuotraukos skirstomos į 2 aplankus: „train“ ir „validation“, kurios naudojamos tinklui treniruoti ir testuoti, mokymui skirta 80 proc. sukauptų nuotraukų, o testavimui – likę 20 proc. Tinklo apmokymui sukurtas programinis kodas pateiktas 33 paveiksle.

```
1 from tensorflow.keras.preprocessing.image import ImageDataGenerator
2 from tensorflow.keras.preprocessing import image
3 from tensorflow.keras.optimizers import Adam
4 import tensorflow as tf
5 import numpy as np
6 import cv2
7
8 #img = cv2.imread("/home/pi/Cropped_Images/Stringing/image2.jpg")
9 #cv2.imshow('image',img)
10
11 train = ImageDataGenerator()
12 validation = ImageDataGenerator()
13
14 train_dataset = train.flow_from_directory ('/home/pi/Cropped_Images/Train/',
15                                           target_size = (200, 200),
16                                           batch_size=10,
17                                           subset = 'training',
18                                           class_mode = 'binary')
19
20 validation_dataset = validation.flow_from_directory ('/home/pi/Cropped_Images/Validation/',
21                                                     target_size = (200, 200),
22                                                     batch_size=10,
23                                                     subset = 'validation',
24                                                     class_mode = 'binary')
25
26 model = tf.keras.models.Sequential([
27     tf.keras.layers.Conv2D(16, (3,3), activation = 'relu', input_shape = (200,200,3)),
28     tf.keras.layers.MaxPool2D(2,2),
29     #
30     tf.keras.layers.Conv2D(32, (3,3), activation = 'relu'),
31     tf.keras.layers.MaxPool2D(2,2),
32     #
33     tf.keras.layers.Conv2D(64, (3,3), activation = 'relu'),
34     tf.keras.layers.MaxPool2D(2,2),
35     #
36     tf.keras.layers.Flatten(),
37     tf.keras.layers.Dense(20, activation = 'relu'),
38     tf.keras.layers.Dropout(0.2),
39     tf.keras.layers.Dense(1, activation = 'sigmoid'),
40 ])
41
42 model.compile(loss = 'binary_crossentropy',
43              optimizer= Adam(
44                  learning_rate=0.0001,
45                  beta_1=0.9,
46                  beta_2=0.999,
47                  epsilon=None,
48                  amsgrad=False,
49                  name="Adam",),
50              metrics = ['accuracy'])
51
52 model_fit = model.fit(
53     #train_dataset, epochs=1)
54     train_dataset,
55     steps_per_epoch =3,
56     epochs = 20
57 )
58 model.save('model.h5')
```

33 pav. Tinklo apmokymui naudojamas programinis kodas

Bandymai atlikti naudojant „RaspberryPi“ 4b mikrovaldiklį, todėl tinklo apmokymui dėl naudojamos operacinės sistemos atminties apribojimo, žingsnis per epochą ir epochų skaičius parinkti maksimalūs. Apmokymo apimties (angl. *batch size*) dydis lygus 10, žingsnis per epochą (angl. *step*

per epoch) – 3, epochų skaičius (angl. *epochs*) – 20. Mokymosi greitis (angl. *learning rate*) lygus 0,0001 žingsnių per iteraciją. Tinklo apmokymas pavaizduotas 34 paveiksle. Pasiektas 86,7 proc. mokymosi tikslumas.

```

pi@raspberrypi:~$ source env/bin/activate
(env) pi@raspberrypi:~$ python tensorflowtest.py
Found 80 images belonging to 2 classes.
Found 0 images belonging to 2 classes.
2021-12-13 22:45:44.298960: W tensorflow/core/framework/cpu_allocator_impl.cc:81 Allocation of 73728 exceeds 10% of system memory.
2021-12-13 22:45:44.299935: W tensorflow/core/framework/cpu_allocator_impl.cc:81 Allocation of 73728 exceeds 10% of system memory.
2021-12-13 22:45:44.300366: W tensorflow/core/framework/cpu_allocator_impl.cc:81 Allocation of 73728 exceeds 10% of system memory.
2021-12-13 22:45:44.371669: W tensorflow/core/framework/cpu_allocator_impl.cc:81 Allocation of 2708480 exceeds 10% of system memory.
2021-12-13 22:45:44.378094: W tensorflow/core/framework/cpu_allocator_impl.cc:81 Allocation of 2708480 exceeds 10% of system memory.
Train for 3 steps
Epoch 1/20
3/3 [=====] - 7s 2s/step - loss: 24.4267 - accuracy: 0.6333
Epoch 2/20
3/3 [=====] - 4s 1s/step - loss: 13.6131 - accuracy: 0.7000
Epoch 3/20
3/3 [=====] - 4s 1s/step - loss: 14.9777 - accuracy: 0.5000
Epoch 4/20
3/3 [=====] - 4s 1s/step - loss: 7.9835 - accuracy: 0.6667
Epoch 5/20
3/3 [=====] - 4s 1s/step - loss: 3.6254 - accuracy: 0.7667
Epoch 6/20
3/3 [=====] - 4s 1s/step - loss: 3.0786 - accuracy: 0.8333
Epoch 7/20
3/3 [=====] - 4s 1s/step - loss: 0.4194 - accuracy: 0.9333
Epoch 8/20
3/3 [=====] - 4s 1s/step - loss: 0.6479 - accuracy: 0.8000
Epoch 9/20
3/3 [=====] - 4s 1s/step - loss: 1.2958 - accuracy: 0.7667
Epoch 10/20
3/3 [=====] - 4s 1s/step - loss: 0.4727 - accuracy: 0.7667
Epoch 11/20
3/3 [=====] - 4s 1s/step - loss: 0.1618 - accuracy: 0.8333
Epoch 12/20
3/3 [=====] - 4s 1s/step - loss: 0.0462 - accuracy: 0.9333
Epoch 13/20
3/3 [=====] - 4s 1s/step - loss: 0.2080 - accuracy: 0.8333
Epoch 14/20
3/3 [=====] - 4s 1s/step - loss: 0.0925 - accuracy: 0.9000
Epoch 15/20
3/3 [=====] - 4s 1s/step - loss: 0.1407 - accuracy: 0.8667
Epoch 16/20
3/3 [=====] - 4s 1s/step - loss: 0.1490 - accuracy: 0.9000
Epoch 17/20
3/3 [=====] - 4s 1s/step - loss: 0.1023 - accuracy: 0.8667
Epoch 18/20
3/3 [=====] - 4s 1s/step - loss: 0.1648 - accuracy: 0.8000
Epoch 19/20
3/3 [=====] - 4s 1s/step - loss: 0.1168 - accuracy: 0.8667
Epoch 20/20
3/3 [=====] - 4s 1s/step - loss: 0.1403 - accuracy: 0.8667
(env) pi@raspberrypi:~$

```

34 pav. Tinklo mokymasis

Modelis išsaugomas „H5“ formatu.

7.6. Apmokyto modelio naudojimas

Apmokytas modelis panaudojamas programiniame kode (žr. 35 pav.), skirtame objekto defektams nustatyti. Apdorojimui pasirenkama nuotrauka, kuri iki tol nebuvo naudojama apmokymo procese.

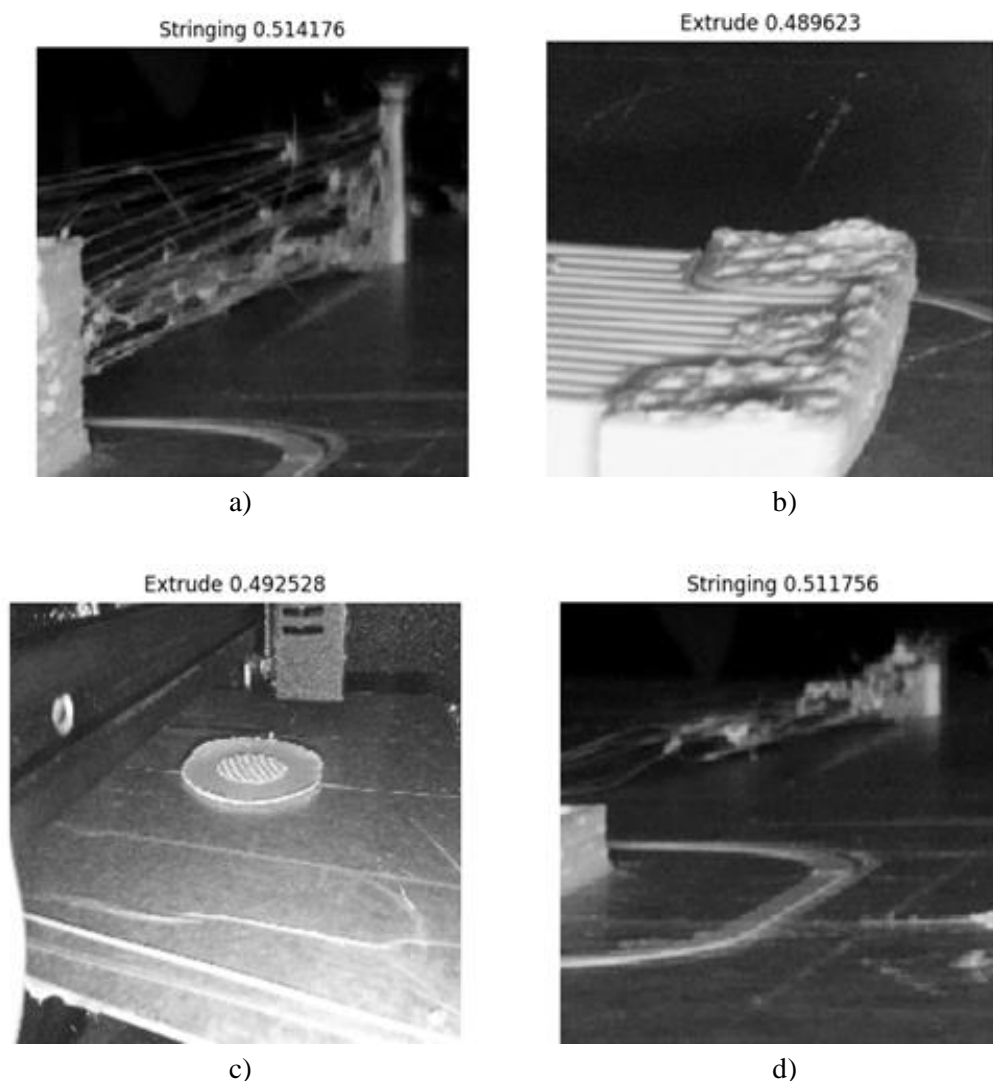
```

1 # load_model_sample.py
2 from tensorflow.keras.models import load_model
3 from tensorflow.keras.preprocessing import image
4 import matplotlib.pyplot as plt
5 import numpy as np
6 from tensorflow import keras
7 import cv2
8
9 model = keras.models.load_model("/home/pi/model.h5")
10 #img_path = '/home/pi/Cropped Images/Train/Stringing/image61.jpg'
11 #img_path = '/home/pi/Cropped Images/Train/OverExtrude/image13.jpg'
12 img_path = '/home/pi/tf/image660.jpg'
13 #img_path = '/home/pi/NotCropped_im/OverExtrude/imageabc.jpg'
14
15
16 img = image.load_img(img_path, target_size=(200, 200))
17 img_tensor = image.img_to_array(img) # (height,
18 a=img_tensor = np.expand_dims(img_tensor, axis=0) # (1, hei
19 img_tensor /= 255. # imshow ex
20
21
22
23 pred = model.predict(a)
24 if pred > 0.5:
25     plt.title('Stringing %1.6f' %pred)
26     plt.imshow(img_tensor[0])
27     plt.axis('off')
28     plt.show()
29
30 else:
31     plt.title('Extrude %1.6f' %pred)
32     plt.imshow(img_tensor[0])
33     plt.axis('off')
34     plt.show()
35

```

35 pav. Programinis kodas objekto defektams nustatyti

Brokui atpažinti naudojamos 200×200 pikselių dydžio nuotraukos. Gaunamas spėjimo tikslumas įrašomas kintamajame „pred“. Jei tikslumas daugiau nei 0,5, brokas priskiriamas defekto tipui „ištempta siūlė“, o jei mažiau nei 0,5 – tipui „išliejamas per didelis medžiagos kiekis“. Gautas rezultatas pavaizduotas 36 paveiksle.



36 pav. Atpažintas brokas nuotraukose: a, d – išsitempusi siūlė; b, c – per didelis išliejamos medžiagos kiekis

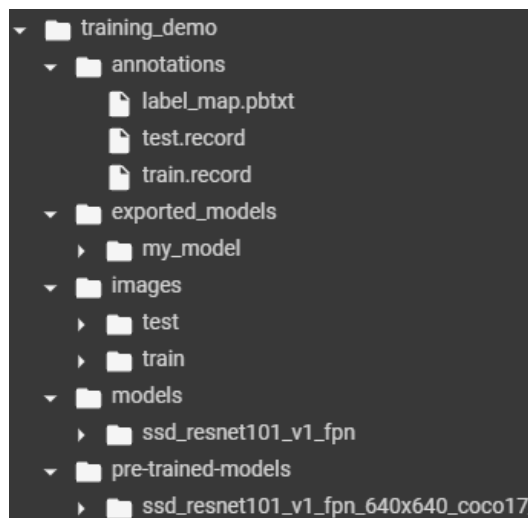
36 paveikslo dalyse informacija nėra tiksli, nes apmokant tinklą be pažymėtų defektų ribų, sudėtinga atskirti ar identifikuotas brokas priskirtas teisingai, matant defektą objekto plote ar kažkokią kitą fone esančią konstrukcinę dalį.

7.7. Broko identifikavimas naudojantis „Google Colab“

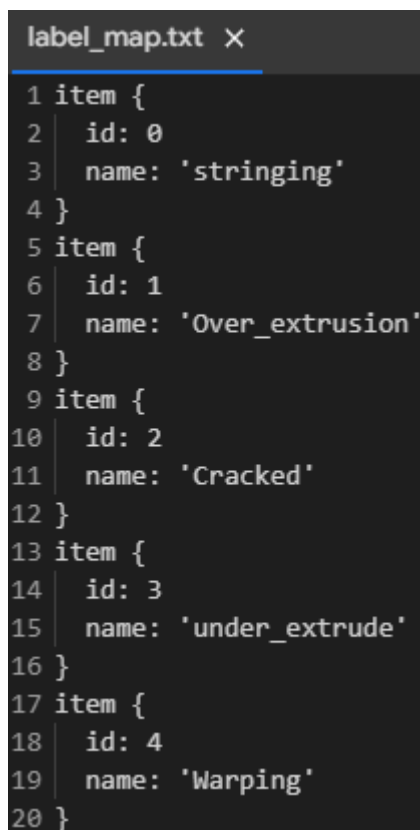
Papildomai atliktas bandymas apmokyti tinklą atpažinti 3D spausdinimo defektus „Google Colaboratory“ aplinkoje, naudojant „Tensorflow“ objektų identifikavimo atvirojo kodo programinės įrangos biblioteką. 37 paveiksle matomas aplankų išdėstymas, tinklui apmokyti. Naudotas apmokyto tinklo „ssd_resnet101_v1“ parametrų failas, kuriame buvo pakeisti tokie nustatymai: žingsnių skaičius per epochą – 1000, tinklo apsimokymo greitis – 0,0004 žingsnių per iteraciją, aprašomos defektų klasės – 5, apmokymo apimtis – 8. Defektų klasės pavaizduotos 38 paveiksle. Gautas

rezultatas apmokant tinklą iki 1000 žingsnių pateiktas 39 paveiksle, o defektų identifikavimo pavyzdys – 40 paveiksle.

Kiekvienam defektui buvo kuriami „XML“ tipo failai, kuriuose aprašytos broko koordinatės. Sekančiame žingsnyje koordinacių rinkinys konvertuojamas į „CSV“ tipo failą. Galiausiai nuotrauka atitinkanti aprašytas koordinatas konvertuojama į „Record“ tipą, kuris naudojamas tinklo mokymuisi.



37 pav. Apmokomo tinklo aplankų išdėstymas



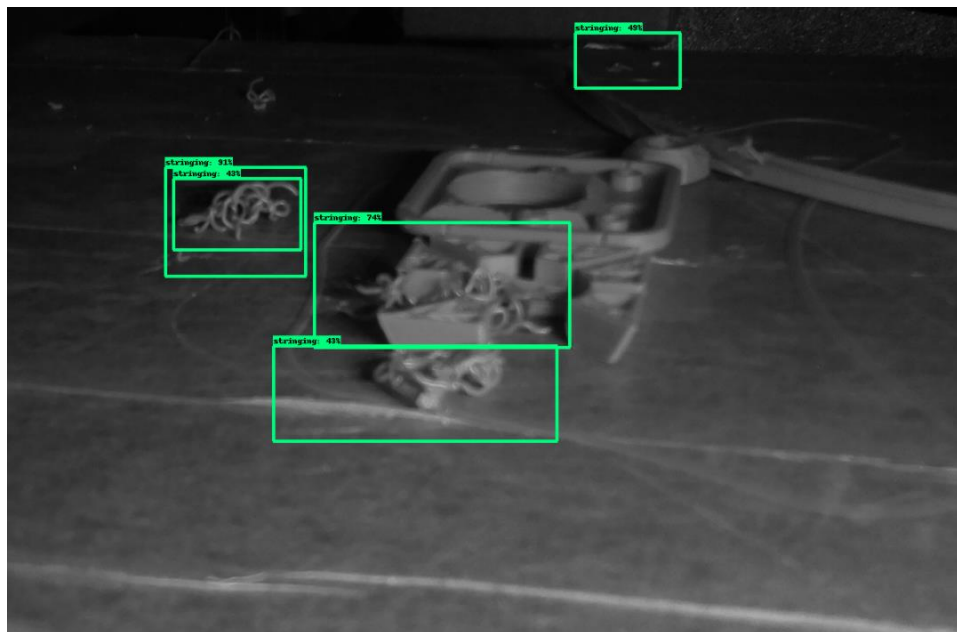
38 pav. Defektų identifikavimo aprašo kodas

```

] INFO:tensorflow:Step 1000 per-step time 3.763s
I1213 19:46:12.424641 140023302104960 model_lib_v2.py:707] Step 1000 per-step time 3.763s
INFO:tensorflow: {'Loss/classification_loss': 0.063123785,
'Loss/localization_loss': 0.020066245,
'Loss/regularization_loss': 0.23806697,
'Loss/total_loss': 0.321257,
'learning_rate': 0.021651587}
I1213 19:46:12.425256 140023302104960 model_lib_v2.py:708] {'Loss/classification_loss': 0.063123785,
'Loss/localization_loss': 0.020066245,
'Loss/regularization_loss': 0.23806697,
'Loss/total_loss': 0.321257,
'learning_rate': 0.021651587}

```

39 pav. Gautas rezultatas apmokant tinklą iki 1000 žingsnių



40 pav. Defektų identifikavimas

Tinklas apmokomas tol, kol pasiekiamas nustatytas epochų ir žingsnių per epochą skaičius. Pasirinktas programos filtras, kurį naudojant nerodomi defektai su mažesniu nei 40 proc. užtikrintumu.

7.8. Parametrų valdymas

Parametrų valdymas atliekamas, naudojantis gautais rezultatais iš apmokyto neuroninio tinklo. Nustatoma broko rūšis, užtikrintumas procentais, nuotraukoje sužymimos ribos pikseliais ir koks iš viso yra aptiktas defektų skaičius.

7.8.1. Žodinis valdymo algoritmas

Pirmame etape tikrinama komunikacija su spausdintuvu:

- jeigu komunikacijos nėra – stabdomas programinis kodas;
- jeigu komunikacijos yra – tikrinama ar gauta komanda fotografuoti.

Kol nėra fotografuojama, apšvietimas iki sekančio tikrinimo išjungtas. Jei komanda fotografuoti gauta:

- įjungiamas apšvietimas;
- įjungiamas kamera;
- naudojant neuroninį tinklą apdorojama nuotrauka;
- nustatytas brokas pažymimas.

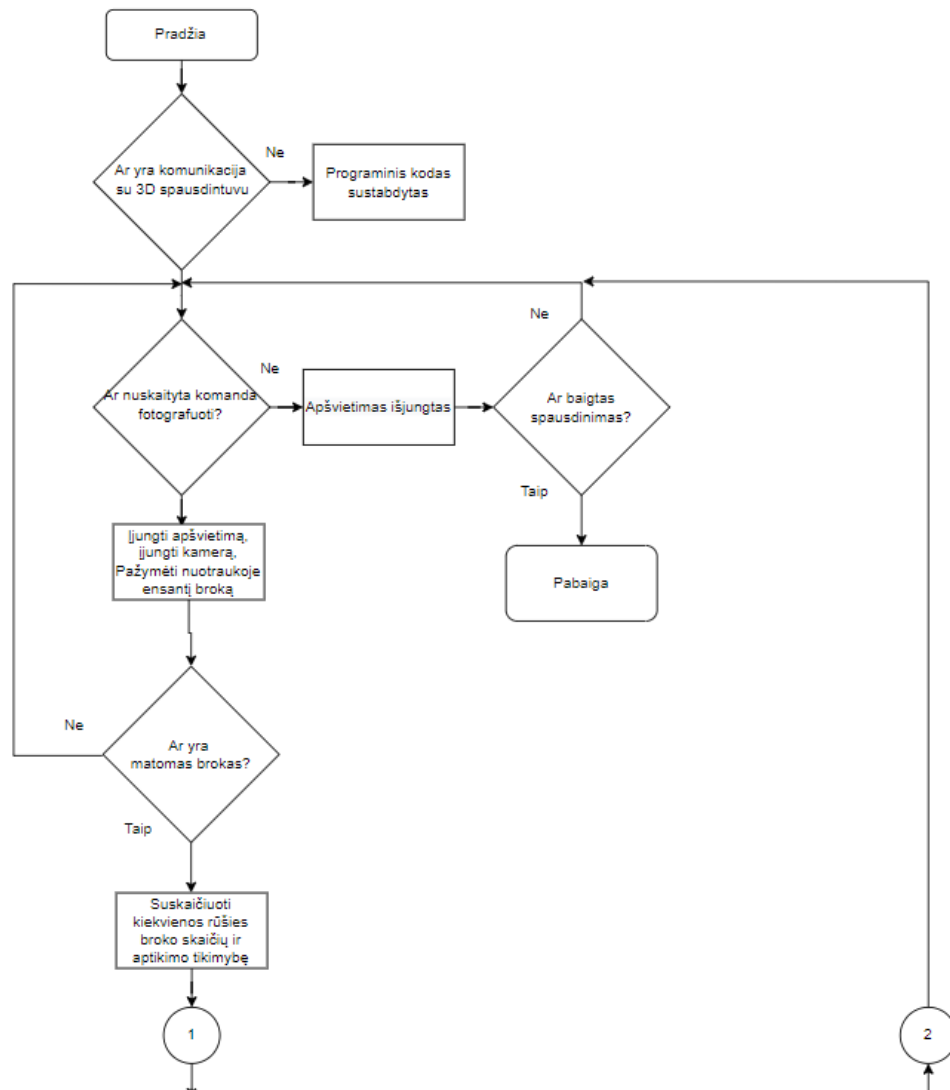
Sekančiame etape nuskaitoma ar yra gautas brokas. Jei taip, skaičiuojamas, kiekvienos rūšies broko skaičius ir aptikimo tikimybė:

- jei daugiausia išsitempusios siūlės broko, tikrinama ar rasti daugiau nei 3 defektai, iš kurių 2 su ne mažesne, nei 60 proc. aptikimo tikimybe, jei taip – programa stabdoma;
- tikrinama kiek yra kitos rūšies defektų, jeigu nėra, grįžtama prie komunikacijos tikrinimo.
- jei daugiausia per didelio medžiagos kiekio broko: tikrinama ar rastas daugiau nei 1 defektas, kuris aptiktas ne mažesne nei 25 proc. tikimybe ir ar medžiagos kiekio parametras yra daugiau nei 100 proc., jei taip, jis sumažinamas iki 100 proc., jei ne – kaitinamo ekstruderio temperatūra sumažinama 5 laipsniais (bet ne mažiau nei 177 °C);
- jei daugiausia per mažo medžiagos kiekio broko, tikrinama ar rastas daugiau nei 1 defektas, kuris aptiktas ne mažesne nei 25 proc. tikimybe ir ar medžiagos kiekio parametras yra mažesnis nei 100 proc., jeigu taip, jis padidinamas iki 100 proc., jei ne – kaitinamo ekstruderio temperatūra padidinama 5 laipsniais (bet ne daugiau nei 250 °C).

Jei aptinkama objekto deformacija (nors 1 defektas), o aptikimo tikimybė yra daugiau nei 10 proc., spausdinimas stabdomas. Jei nustatoma, kad objektas įskilęs (atsiskyre sluoksniai, nors 1 defektas), o aptikimo tikimybė daugiau nei 25 proc., spausdinimas stabdomas.

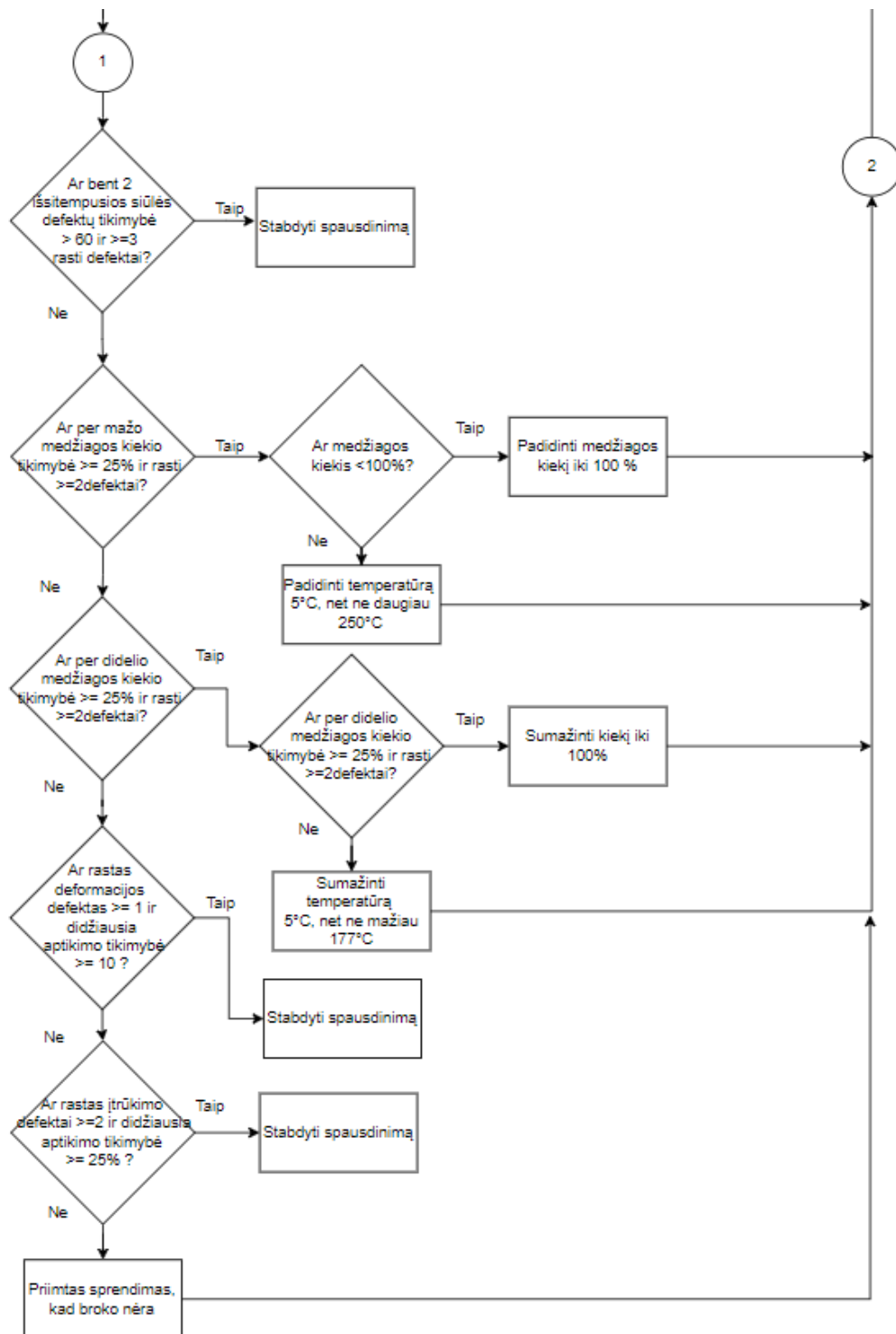
7.8.2. Struktūrinis valdymo algoritmas

41 ir 42 paveiksluose pavaizduotos sistemos valdymo algoritmo dalys.



41 pav. Struktūrinis valdymo algoritmas (1)

Pirmojoje algoritmo dalyje, 41 paveiksle, parodyta kaip vykdomas komunikacijos su 3D spausdintuvu išlaikymas, kameros bei apšvietimo valdymas. Priimamas sprendimas ar matomas brokas, ir koks jis yra.



42 pav. Struktūrinis valdymo algoritmas (2)

Antroje algoritmo dalyje, 42 paveiksle, pavaizduotas sprendimo priėmimas, esant tam tikrai broko rūšiai.

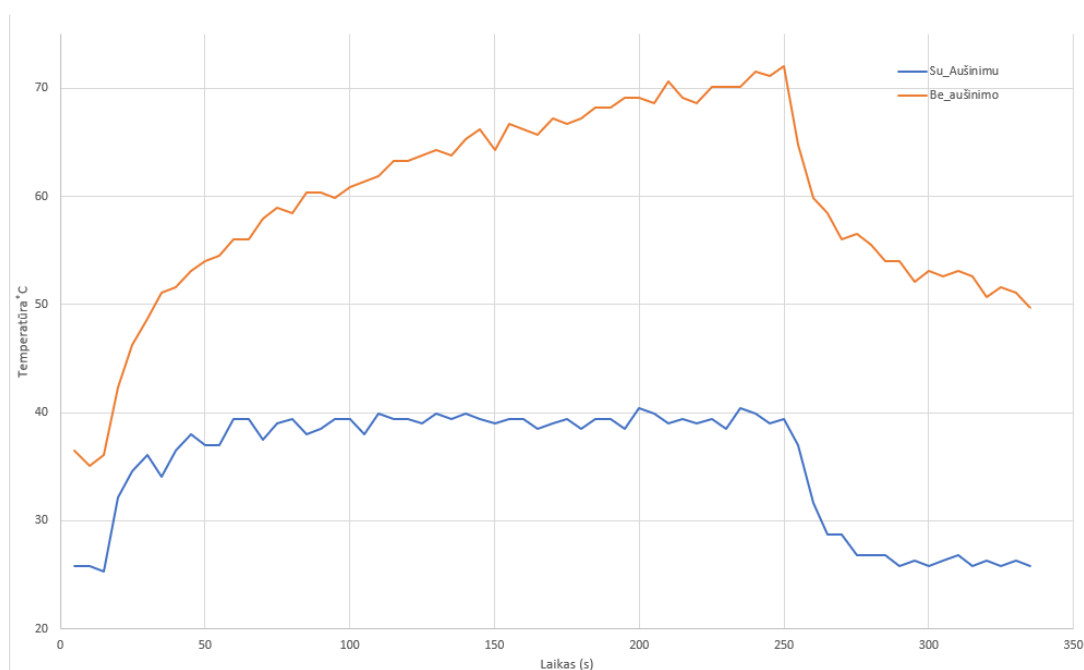
8. Tiriamoji dalis

Šioje dalyje atliekami centrinio procesoriaus temperatūros valdymo, ekstruderio temperatūros valdymo pasinaudojant neuroninius tinklus vaizdo atpažinime tyrimas. Ir spalvoto apšvietimo su skirtingų spalvų detalėmis tyrimas.

8.1. „Raspberry Pi“ mikrokompiuterio centrinio procesoriaus temperatūros valdymas

Naudojant vaizdo apdorojimo programinius kodus, pastebėta, kad mikrokompiuteris keletą kartų išsijungė ir po kelių minučių vėl įsijungė. „RaspberryPi“ mikrokompiuteryje integruota komanda „vcgencmd measure_temp“ nustatyta, kad procesoriaus temperatūra viršijo 60 °C.

Atliktas bandymas, paleidžiant vaizdo atpažinimo programinį kodą ir stebint temperatūros pokyčius. 43 paveiksle matyti, kad be aušinimo per 4 minutes temperatūra pakilo iki 71,5 °C. Kadangi aukštesnė temperatūra yra pavojinga mikrokompiuterio procesoriui, išjungtas vaizdo atpažinimas. Norint apsaugoti procesorių nuo perkaitimo reikalingas aušinimas. Pasirinktas 5 V, 0,2 A nešiojamojo kompiuterio ventiliatorius, kuriam atspausdintas oro srauto nukreipimo laikiklis. Užtikrinus procesoriaus aušinimą, jo temperatūra paleidus vaizdo atpažinimo kodą per 4 minutes nepakilo daugiau nei iki 41 °C, o išjungus programinį kodą, stabilizavosi ties 26 °C.



43 pav. „RaspberryPi“ CPU temperatūra su ir be aušinimo

Ventiliatorius valdomas *GPIO* kontaktu, kuris įjungia ventiliatorių, pasiekus 35 °C temperatūrą.

8.2. 3D spausdintuvo parametrų valdymas

Tiriama 3D spausdinimo sistemos reakcija ir sprendimų priėmimas spausdinant 30 mm × 30 mm objektą iš „PLA“ plastiko. Kaitinamo ekstruderio skylės skersmuo 0,4 mm.

Brokas žymimas tuomet, jeigu nustatyta didesnė nei 20 proc. tikimybė.

8.2.1. 1 bandymas

Nustatomi parametrai per didelio medžiagos kiekio brokui išgauti:

- kaitinamo ekstruderio temperatūra – 240 °C;
- kaitinamos pagrindo plokštės temperatūra – 20 °C;
- išspaudžiamos medžiagos kiekis – 120 proc.

44 paveiksle pavaizduotas rastas 2 rūšių brokas: „stringing“ – išsitempusi siūlė – 43 proc., „over_extrusion“ – per didelis medžiagos kiekis – 21 ir 26 procentai.

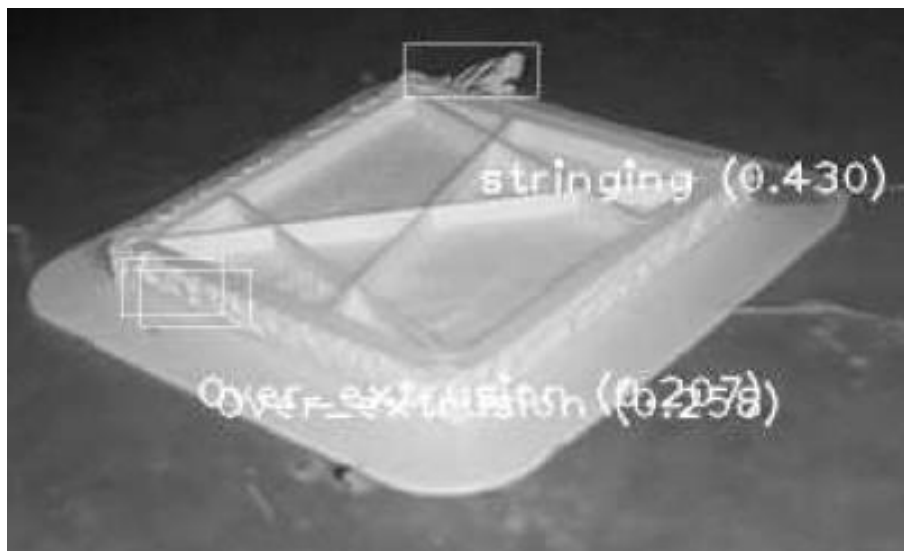


image 568, Kiekis= 3, Rūšis= 1, Didžiausias_%= 43, Sprendimas= Nera, Temperatura= 234.7, KartBrokas= 2

44 pav. 1 bandymo rezultatas

Rezultatas. Rasti 2 rūšių defektai yra skirtingo pobūdžio, todėl nėra tenkinama sąlyga parametrams keisti. Priimamas sprendimas – broko nėra.

8.2.2. 2 bandymas

Iki 20 proc. sumažinama sistemos defekto nustatymo tikimybė. 45 paveiksle pavaizduoti rasti 4 defektai, iš kurių 3 yra pasikartojantys: „stringing“ – išsitempusi siūlė – 69 proc., „over_extrusion“ – per didelis medžiagos kiekis – 21, 23 ir 23 procentai.

Rezultatas. Rasti 4 defektai, o iš jų 3 pasikartojantys, todėl tenkinama sąlyga parametrams keisti. Priimamas sprendimas – „M5“, t. y. 5 °C sumažinama kaitinamo ekstruderio temperatūra.

46 paveiksle pavaizduoti bandymo metu kaupiami duomenys: nuotraukų skaičius, rasto broko kiekis, defekto rūšis, didžiausia nustatyto broko tikimybė, priimtas sprendimas, kaitinamo ekstruderio temperatūra bei vienodos rūšies defektų skaičius.

Iš 46 paveikslo matyti, kad neuroninis tinklas geriausiai apmokytas atpažinti išsitempusios siūlės defektus, atpažinimo tikimybė siekia iki 73 procentų.

47 paveiksle pavaizduotas kaitinamo ekstruderio temperatūros pokytis, priimant sprendimus, po kiekvieno sluoksnio nuotraukos analizės. „image 577“ atitinka 29 sluoksnį grafike.

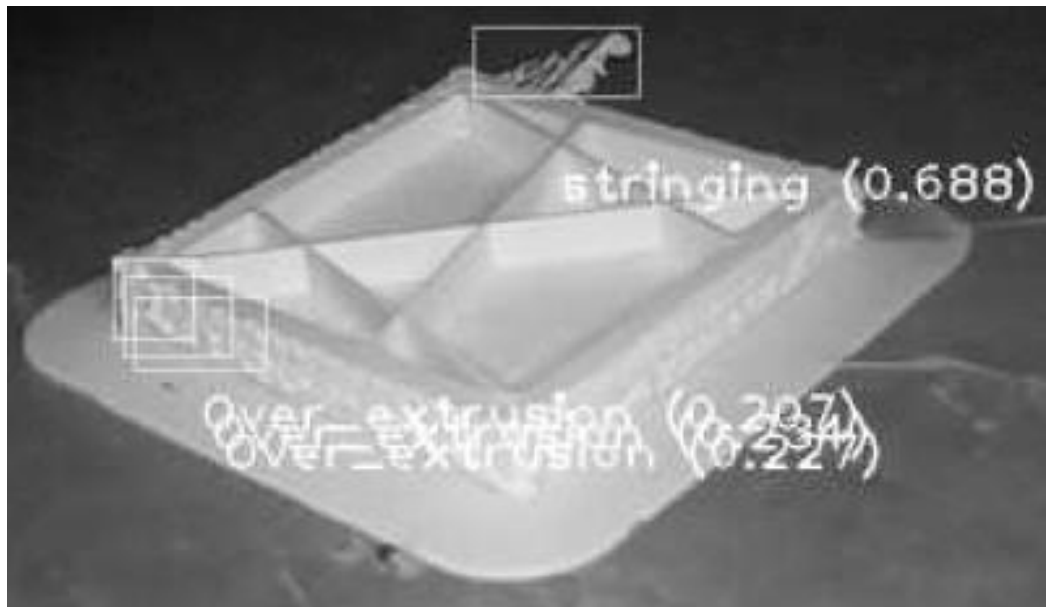


image 577, Kiekis= 4, Ruisis= 1, Didžiausias_%= 68, Sprendimas= M5, Temperatura= 234.9, KartBrokas= 3

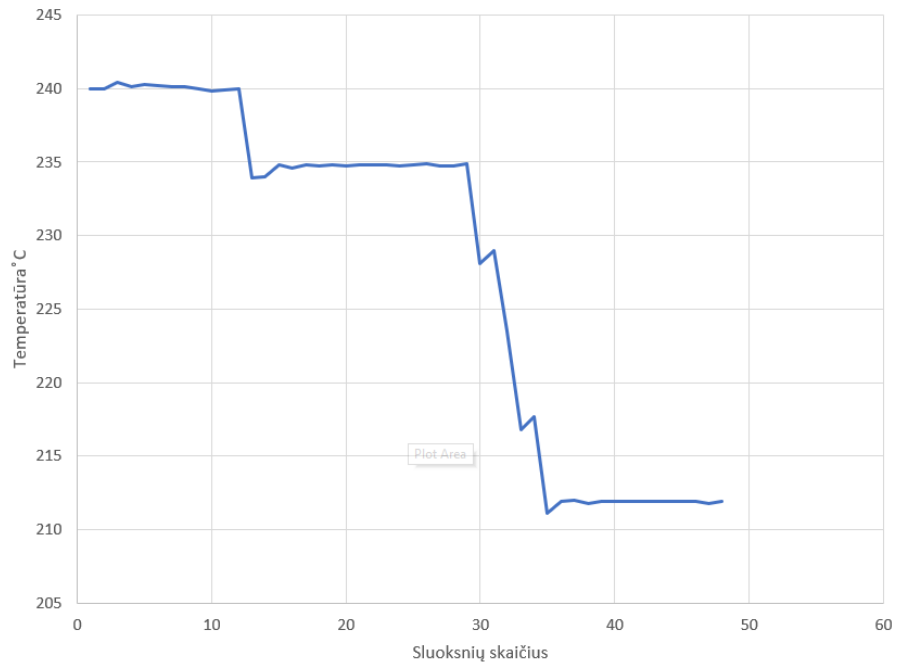
45 pav. 2 bandymo rezultatas

image 570, Kiekis= 2, Ruisis= 0, Didžiausias_%= 55, Sprendimas= Nera, Temperatura= 234.8, KartBrokas= 1
 image 571, Kiekis= 1, Ruisis= 0, Didžiausias_%= 55, Sprendimas= Nera, Temperatura= 234.8, KartBrokas= 1
 image 572, Kiekis= 2, Ruisis= 0, Didžiausias_%= 64, Sprendimas= Nera, Temperatura= 234.7, KartBrokas= 1
 image 573, Kiekis= 1, Ruisis= 0, Didžiausias_%= 59, Sprendimas= Nera, Temperatura= 234.8, KartBrokas= 1
 image 574, Kiekis= 2, Ruisis= 0, Didžiausias_%= 67, Sprendimas= Nera, Temperatura= 234.9, KartBrokas= 1
 image 575, Kiekis= 3, Ruisis= 1, Didžiausias_%= 67, Sprendimas= Nera, Temperatura= 234.7, KartBrokas= 2
 image 576, Kiekis= 2, Ruisis= 0, Didžiausias_%= 60, Sprendimas= Nera, Temperatura= 234.7, KartBrokas= 1
 image 577, Kiekis= 4, Ruisis= 1, Didžiausias_%= 68, Sprendimas= M5, Temperatura= 234.9, KartBrokas= 3
 image 578, Kiekis= 3, Ruisis= 1, Didžiausias_%= 64, Sprendimas= Nera, Temperatura= 228.1, KartBrokas= 2
 image 579, Kiekis= 5, Ruisis= 1, Didžiausias_%= 69, Sprendimas= M5, Temperatura= 229.0, KartBrokas= 4
 image 580, Kiekis= 4, Ruisis= 1, Didžiausias_%= 69, Sprendimas= M5, Temperatura= 223.5, KartBrokas= 3
 image 581, Kiekis= 2, Ruisis= 0, Didžiausias_%= 73, Sprendimas= Nera, Temperatura= 216.8, KartBrokas= 1
 image 582, Kiekis= 4, Ruisis= 1, Didžiausias_%= 73, Sprendimas= M5, Temperatura= 217.7, KartBrokas= 3
 image 583, Kiekis= 3, Ruisis= 1, Didžiausias_%= 73, Sprendimas= Nera, Temperatura= 211.1, KartBrokas= 2
 image 584, Kiekis= 1, Ruisis= 0, Didžiausias_%= 72, Sprendimas= Nera, Temperatura= 211.9, KartBrokas= 1
 image 585, Kiekis= 2, Ruisis= 0, Didžiausias_%= 69, Sprendimas= Nera, Temperatura= 212.0, KartBrokas= 1
 image 586, Kiekis= 3, Ruisis= 1, Didžiausias_%= 59, Sprendimas= Nera, Temperatura= 211.8, KartBrokas= 2
 image 587, Kiekis= 2, Ruisis= 0, Didžiausias_%= 59, Sprendimas= Nera, Temperatura= 211.9, KartBrokas= 1
 image 588, Kiekis= 2, Ruisis= 0, Didžiausias_%= 55, Sprendimas= Nera, Temperatura= 211.9, KartBrokas= 1
 image 589, Kiekis= 2, Ruisis= 0, Didžiausias_%= 58, Sprendimas= Nera, Temperatura= 211.9, KartBrokas= 1
 image 590, Kiekis= 2, Ruisis= 0, Didžiausias_%= 54, Sprendimas= Nera, Temperatura= 211.9, KartBrokas= 1

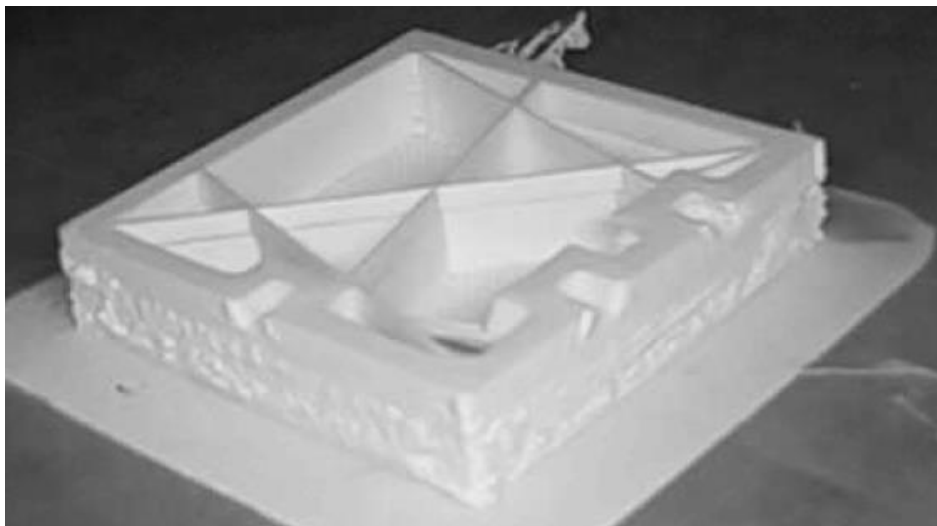
46 pav. Bandymo metu sukaupti duomenys

Objektas spausdinamas su 5 proc. užpildu, todėl ne visada iki kol padaroma sekanti nuotrauka spėja nusistovėti temperatūros pokytis.

Atliktame bandyme kaitinamo ekstruderio temperatūra nusistovėjo ties 212 °C. Spausdinamo gaminio dalis po temperatūros nusistovėjimo priėmus sprendimą (žr. 48 pav.) sumažinti temperatūrą.



47 pav. Kaitinamo ekstruderio temperatūros pokytis priimant sprendimus (1)



48 pav. Objekto paviršius keičiantis temperatūrai

Naujame sluoksnyje nebesimato perliejamos medžiagos kiekio, gaminio paviršius tampa lygus.

8.2.3. 3 bandymas

Nustatomi parametrai per mažo medžiagos kiekio brokui išgauti:

- kaitinamo ekstruderio temperatūra – 180 °C;
- kaitinamos pagrindo plokštės temperatūra – 20 °C;
- išspaudžiamos medžiagos kiekis – 50 proc.

49 paveiksle pavaizduotas rastas 1 rūšies brokas: „under_extrude“ – per mažas medžiagos kiekis – 61, 25 ir 22 proc.

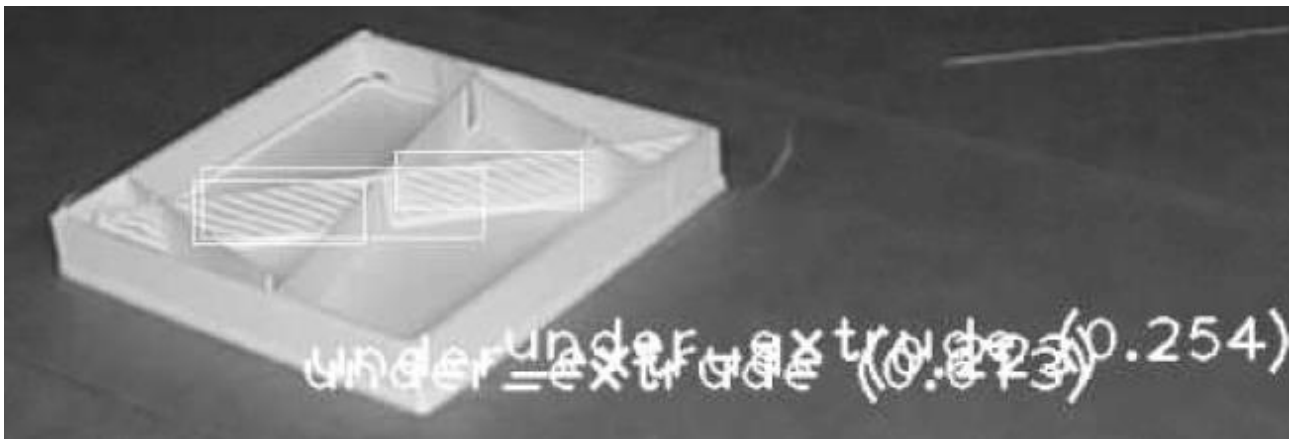


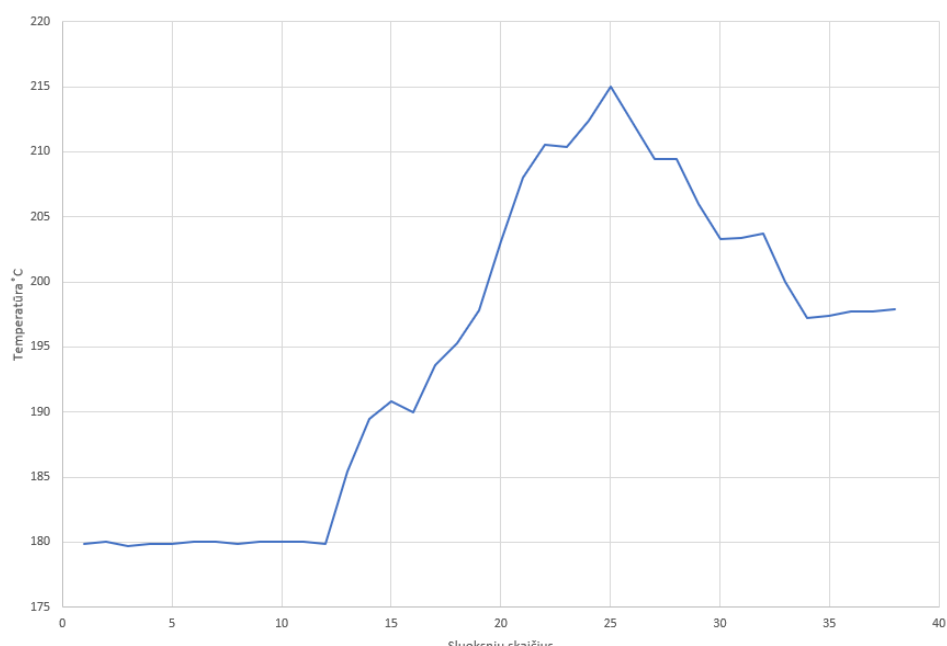
image 740, Kiekis= 3, Ruis= 3, Didžiausias_%= 61, Sprendimas= P5, Temperatura= 179.9, KartBrokas= 3

49 pav. 3 bandymo rezultatas

Rezultatas. Rastas 1 rūšies defektas yra pasikartojantis 3 kartus, todėl tenkinama sąlyga parametrams keisti. Priimamas sprendimas – „P5“, t. y. 5 °C padidinama kaitinamo ekstruderio temperatūra.

50 paveiksle pavaizduotas kaitinamo ekstruderio temperatūros pokytis, priimant sprendimus po kiekvieno sluoksnio nuotraukos analizės. „image 740“ atitinka 11 sluoksnį grafike. Objektas spausdinamas su 5 proc. užpildu, todėl ne visada iki kol padaroma sekanti nuotrauka spėja nusistovėti temperatūros pokytis.

Grafike (žr. 50 pav.) matomas temperatūros kilimas iki 25 sluoksnio, kai kaitinamo ekstruderio temperatūra pasiekia 215 °C. Nors brokas nebuvo ištaisytas, sistema priėmė sprendimą mažinti temperatūrą. Peržiūrėjus nuotraukas, pastebėta, kad žymimi defektai nebuvo tinkamai nustatyti (žr. 51 pav.).



50 pav. Kaitinamo ekstruderio temperatūros pokytis priimant sprendimus (2)

Grafike matyti, kad neteisingas sprendimas buvo priimtas 3 kartus.

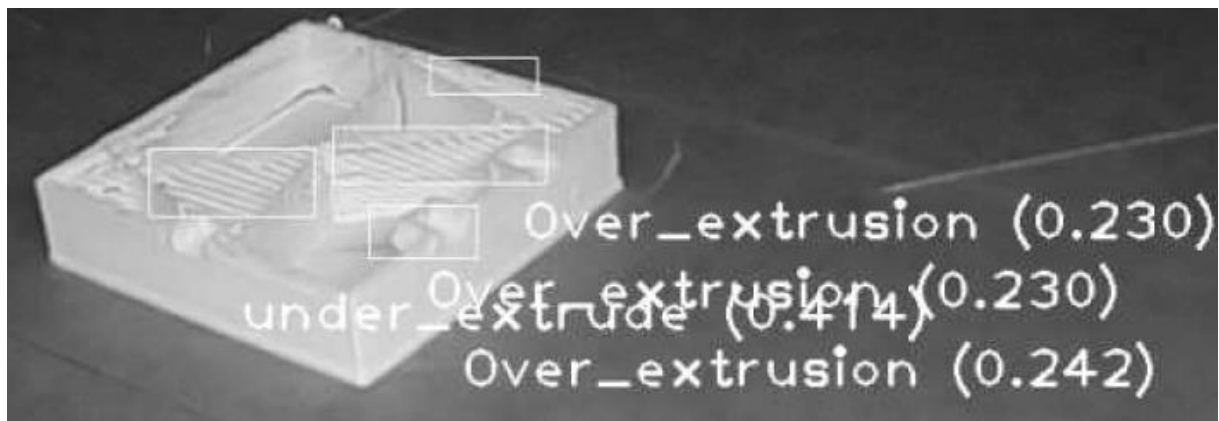


image 753, Kiekis= 4, Ruisis= 1, Didžiausias_%= 41, Sprendimas= M5, Temperatura= 215.0, KartBrokas= 3

51 pav. Nustatytas neteisingas defektas

Iš 4 rastų defektų vienas – per didelis medžiagos kiekis – pažymėtas neteisingai (23 proc. tikimybė), todėl priimtas sprendimas sumažinti kaitinamo ekstruderio temperatūrą, kai to daryti nereikėtų.

8.2.4. 4 bandymas

Sistemos parametrai, objektą spausdinant iš „ABS“ plastiko:

- kaitinamo ekstruderio temperatūra – 240 °C;
- kaitinamos pagrindo plokštės temperatūra – 45 °C;
- išspaudžiamos medžiagos kiekis – 100 proc.

Bandyme naudojant nurodytus parametrus ir objektą spausdinant iš „ABS“ plastiko, bandoma išgauti tokias defekto rūšis: detalės skilinėjimą arba deformacijos susidarymą.

52 paveiksle pavaizduotas broko nustatymo procesas, kai atpažinimo tikimybė siekia 25 proc. Nors objekto kampai atlipo nuo pagrindo, defektas nebuvo rastas. Sistema priėmė, kad detalė nėra brokuota ir tęsė spausdinimą.



image 1640, Kiekis= 0, Ruisis= 0, Didžiausias_%= 0, Sprendimas= Nera,

52 pav. Neatpažintas deformacijos brokas

Pastebėjus, kad nėra tinkamai priimtas valdymo sprendimas, atpažinimo tikimybė sumažinta iki 10 procentų. Siekiama nustatyti ar neuroninis tinklas apsimokė ir geba atpažinti deformuotus defektus.

53 paveiksle matyti, kad „Warping“ deformacijos brokas nustatomas su 10,9 proc. tikimybe.



image 1641, Kiekis= 5, Ruisis= 0, Didžiausias_%= 23, Sprendimas= Stop,

53 pav. Nustatytas deformacijos brokas

Sumažinus broko atpažinimo tikimybę taip pat matyti ir neteisingai sužymėti defektai. Sistema atpažino „stringing“ defektą – išsitempusią siūlę – su 11 ir 14 proc., tikimybėmis kurios pažymi atlipusią juostą, nors tikrasis spausdinimo brokas – „Warping“ – objekto deformacija, kurio kampas atlipo nuo pagrindo.

8.3. Algoritmo optimizavimas

Atliekant bandymus pastebėta, kad nuotraukose, kuriose pažymėtos kelios defektų rūšys (žr. 54 pav.), nėra priimami teisingi sprendimai.

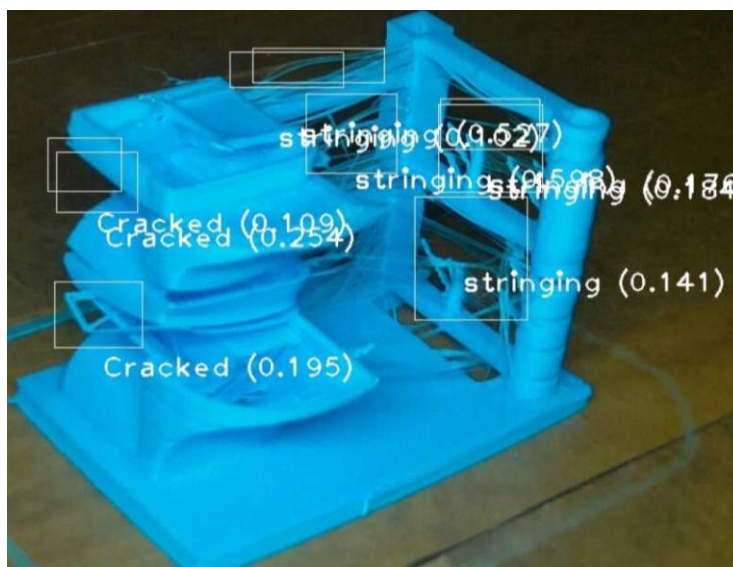


image 1830, Kiekis= 9, Ruisis= 0, Didžiausias_%= 59, Sprendimas= stop,

54 pav. Vizualizuojami aptikti defektai

Esant išsitempusios siūlės defektui su didžiausia atpažinimo tikimybe ir keliems kitos pat rūšies defektams, priimamas sprendimas nieko valdymo programoje nekeisti.

Kadangi kiekvienos rūšies broko atpažinimo sistema apmokyta ne vienodai, neuroninis tinklas atpažįsta išsitempusios siūlės defektus su 60–70 proc. tikimybe, deformacijos defektus – su 10–15 proc. tikimybe, įskilimų defektus – su 20–30 proc. tikimybe. Todėl nuspręsta tikrinti defektų kiekį ir atpažinimo tikimybę kiekvienai rūšiai. Iš gautų rezultatų nustatyta kiekvienos rūšies didžiausia aptikimo tikimybė ir jeigu nustatyta riba viršijama, priimamas sprendimas koreguoti parametrus arba stabdyti spausdinimą.

Naudojant šį būdą vizualizacijoje padaugėja nustatytų defektų skaičius, tačiau jie neįtakoja sprendimo priėmimo, nes brokas atrenkamas pagal individualios grupės defekto aptikimo tikimybę.

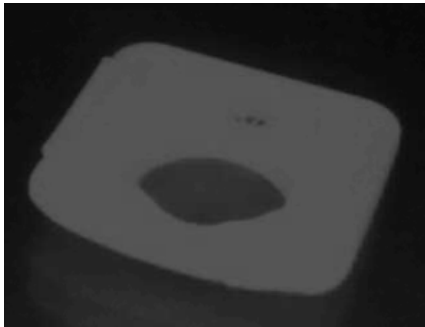
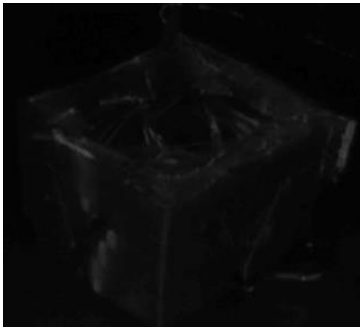
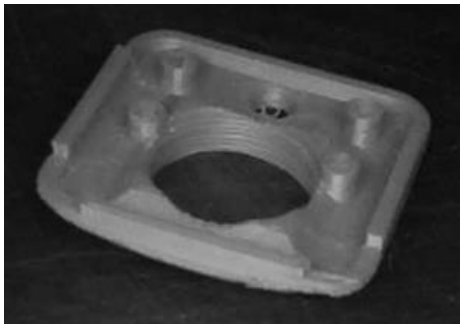
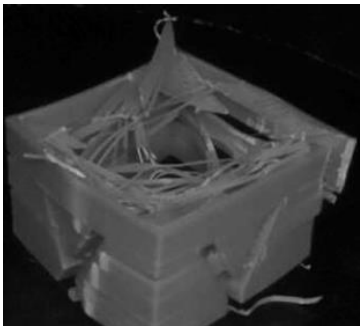
Rezultatas: pagal naują algoritmą, išsitempusios siūlės defektas nesiekė 60 proc., todėl spausdinimo stabdymas įvyko dėl rasto kito defekto – įskilimo, kurio didžiausia atpažinimo tikimybė 25 procentai.

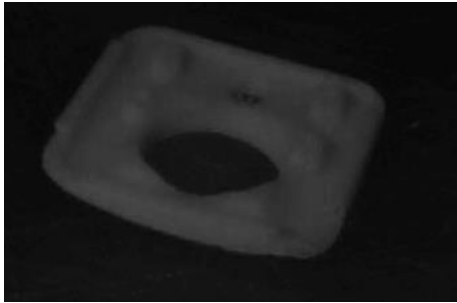
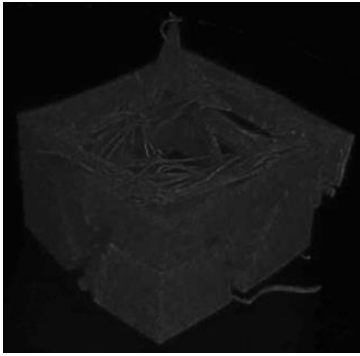
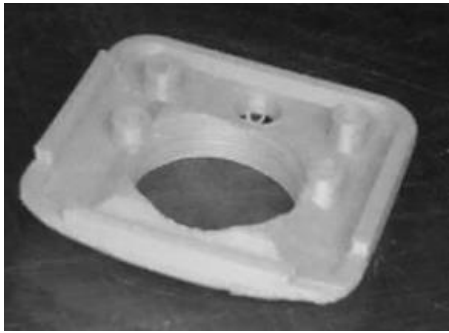
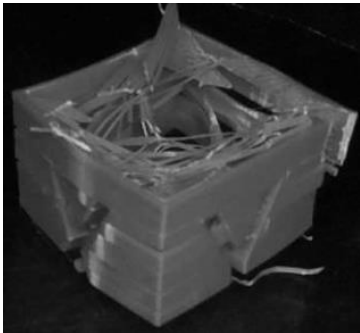
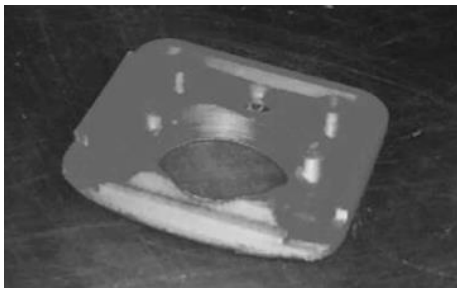
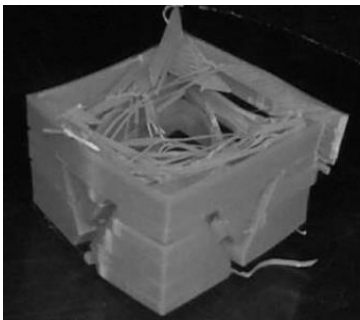
8.4. Spalvoto apšvietimo bandymas

Naudojant rožinį ar raudoną spausdinimo plastikus ir vykdant defektų apmokymą bei atpažinimą, susidurta su problema. Fiksuojamuose nuotraukose pastebėta, kad esant baltos spalvos šviesos šaltiniui, nėra matomos spausdinamo objekto linijos, o tik jo vientisas kontūras. Dėl ko negalima identifikuoti broko detalės paviršiuje.

3 lentelėje pavaizduoti eksperimentinių bandymų rezultatai, t. y. pirmame stulpelyje rožinės spalvos, o antrame stulpelyje mėlynos spalvos objektų nuotraukos, esant skirtingų spalvų apšvietimui: raudonam, žaliai, mėlynai, geltonai, baltai.

3 lentelė. Objektų paviršių nuotraukos apšviečiant skirtingos spalvos šviesa (1)

Apšvietimo spalva	Originali objekto spalva: rožinė	Originali objekto spalva: mėlyna
Raudona		
Žalia		

Apšvietimo spalva	Originali objekto spalva: rožinė	Originali objekto spalva: mėlyna
Mėlyna		
Geltona		
Balta		

Rožinės spalvos objekto paviršiaus analizė, kai naudojama apšvietimo spalva:

- raudona: matomas vientisas objekto kontūras, nors jis sudarytas iš siūlių, nematomos jokios paviršiuje esančios struktūros;
- žalia: matomas objekto kontūras ir visos jo paviršiuje esančios dalys, nėra vaizdo susiliejimo ar atspindžių;
- mėlyna: neryškiai matomos paviršiuje esančios objekto dalys, detalė tamsi;
- geltona: matomas objekto kontūras ir visos jo paviršiuje esančios dalys, nėra vaizdo susiliejimo ar atspindžių;
- balta: matomas vientisas objekto kontūras, neryškiai matomos paviršiuje esančios struktūros.

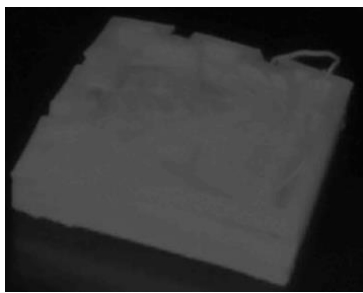
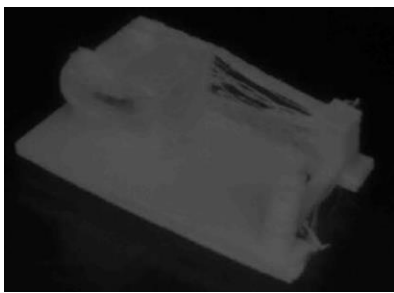

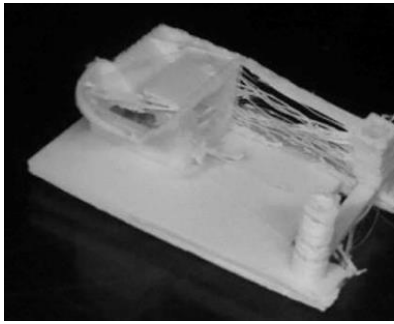
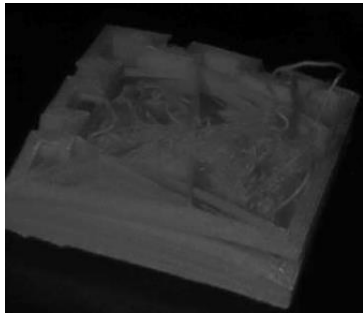
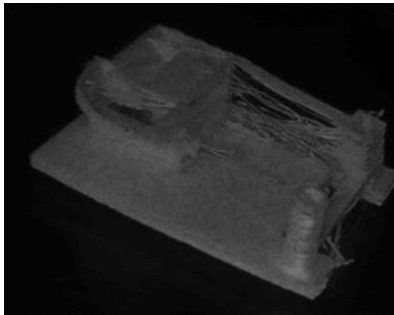

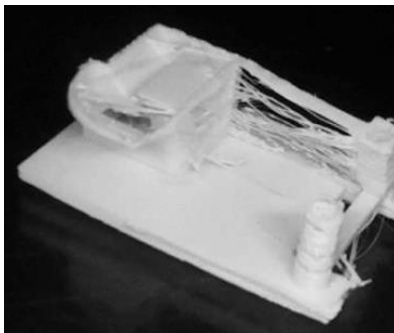
Mėlynos spalvos objekto paviršiaus analizė, kai naudojama apšvietimo spalva:


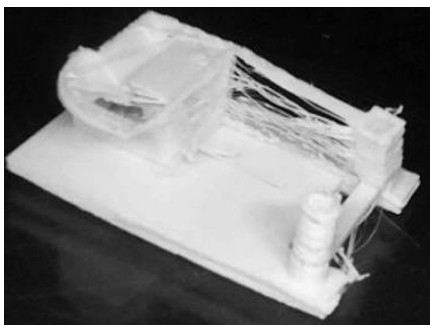
- raudona: objektas nematomas;
- žalia: objekto paviršiuje matomi įtrūkimai, o vidinėje jo dalyje – išsitempusios siūlės brokas;
- mėlyna: matomas neryškiai tamsiame fone išsiskiriantis objektas, paviršiuje esantys įtrūkimai nėra matomi;
- geltona: objekto paviršiuje matomi įtrūkimai, o vidinėje jo dalyje – išsitempusios siūlės brokas;

- balta: objekto paviršiuje matomi įtrūkimai, o vidinėje jo dalyje – išsitempusios siūlės brokas.

4 lentelėje pavaizduoti eksperimentinių bandymų rezultatai, t. y. pirmame stulpelyje geltonos spalvos, o antrame stulpelyje baltos spalvos objektų nuotraukos, esant skirtingų spalvų apšvietimui: raudonam, žalia, mėlynam, geltonam, baltam.

4 lentelė. Objektų paviršių nuotraukos apšviečiant skirtingos spalvos šviesa (2)

Apšvietimo spalva	Originali objekto spalva: geltona	Originali objekto spalva: balta
Raudona		
Žalia		
Mėlyna		
Geltona		

Apšvietimo spalva	Originali objekto spalva: geltona	Originali objekto spalva: balta
Balta		

Geltonos spalvos objekto paviršiaus analizė, kai naudojama apšvietimo spalva:

- raudona: objekto kontūro, kuriame matomi neryškūs išsitempusios siūlės defektai, paviršius susiliejęs, nėra matomas įtrūkimas;
- žalia: objekto paviršiuje ryškiai matomas įtrūkimas, o vidinėje jo dalyje – išsitempusios siūlės brokas;
- mėlyna: objektas neryškiai išsiskiria iš tamsaus fono, jo paviršiuje esantis įtrūkimas nėra matomas, o vidinėje dalyje negalima identifikuoti broko;
- geltona: objekto paviršiuje ryškiai matomas įtrūkimas, o vidinėje jo dalyje – išsitempusios siūlės brokas;
- balta: objekto paviršiuje ryškiai matomas įtrūkimas, o vidinėje jo dalyje – išsitempusios siūlės brokas.

Baltos spalvos objekto paviršiaus analizė, kai naudojama apšvietimo spalva:

- raudona: objekto kontūro, kuriame matomi neryškūs išsitempusios siūlės defektai, paviršius susiliejęs;
- žalia: ryškiai matomas išsitempusios siūlės brokas, nėra atspindžių ar vaizdo susiliejimo;
- mėlyna: objektas neryškiai išsiskiria iš tamsaus fono, paviršiuje esančios jo dalys nėra matomos, todėl negalima identifikuoti broko;
- geltona: ryškiai matomas išsitempusios siūlės brokas, nėra atspindžių ar vaizdo susiliejimo;
- balta: ryškiai matomas išsitempusios siūlės brokas, nėra atspindžių ar vaizdo susiliejimo.

Rezultatas. Iš 4 naudojamų spausdinamo plastiko spalvų ryškiausiai nuotraukose matomi defektai naudojant žalios šviesos šaltinį. Geltona ir balta šviesa apšvietus tiriamą objektą, brokas ir paviršiuje esančios struktūros gerai matomos kai detalėms spausdinti naudojamas mėlynos, baltos ir geltonos spalvų plastikai. Naudojant mėlynos spalvos šviesos šaltinį, objektas neryškiai išsiskiria iš tamsaus fono ir nėra matomi jo defektai. Esant raudonos spalvos apšvietimui, visuose bandymuose objekto paviršius spalva buvo vientisa, o defektai ar kitos struktūros nebuvo matomos.

Išvados

1. Išnagrinėti 3 praktikoje naudojami 3D spausdinimo metodai: liejant siūlą ant pagrindo (FDM), stereolitografika (SLA) bei spausdinimas metalo milteliais. Tyrimui atlikti pasirinktas FDM spausdinimo metodas, nes liejamos medžiagos brokas gali būti aptiktas naudojant vaizdo kamerą. Techninės literatūros apžvalga parodė, kad tai vienas pigiausių rinkoje esančių 3D spausdinimo metodų.
2. Apibrėžtos pagrindinės 3D spausdinimo broko rūšys, iš kurių vaizdui atpažinti pasirinkti: nesuderintas pagrindo plokštės aukštis, sukibimo su pagrindo plokšte trūkumas, išliejamas per didelis ar per mažas medžiagos kiekis, sluoksnių atsiskyrimas (skilinėjimas). Visas pasirinktas 5-ias broko rūšis galima tiesiogiai atpažinti naudojant vaizdo kamerą.
3. Tyrimai parodė, kad 3D spausdintų objektų defektams atpažinti gali būti naudojama tiek viena, tiek keletas kamerų. Sistemose su daugiau nei viena kamera brokas nustatomas, palyginant modelio ir gauto vaizdo geometriją, kuri turi sutapti. Šio projekto eksperimentinėje sistemoje sumontuota 1 vaizdo kamera, gebanti matyti 3 objekto paviršius.
4. Išnagrinėti trys dirbtinio neuroninio tinklo apmokymo metodai: prižiūrimas, neprižiūrimas ir sustiprintas. Tyrime naudotas prižiūrimasis mokymasis, kuris leidžia išskirti broko grupes.
5. Aprašyti pagrindiniai vaizdo atpažinimo kameros kalibravimo būdai: automatinis, neplaninio objekto ir naudojant šaškių lentą. Tiriamosioje sistemoje kamera nebuvo kalibruota, kadangi pasirinktoms broko rūšims atpažinti nereikia geometrinio tikslumo.
6. Pasirinkti tokie defektų stebėjimo įrangos komponentai: mikrokompiuteris „Raspberry Pi“ 4b, kuriame kaupiamos nuotraukos ir palaikoma komunikacija su 3D spausdintuvu, „Raspberry Noir“ V2 8MP kamera tinkanti naudoti su mikrokompiuteriu. Naudota tokia programinė įranga: „Raspbian Debian“ 10, 32-bitų operacinė sistema, „Tensorflow“ 2.0.0 programa.
7. Sukurtas nuotraukų kaupimo ir broko tikrinimo algoritmas, užtikrinantis komunikaciją su 3D spausdintuvu, nuotraukų apdorojimą ir priimamus sprendimus defektams ištaisyti. Vaizdai apdorojami ir pažymimi 2 kadru per sekundę greičiu esant 1280×720 pikselių nuotraukų rezoliucijai.
8. Sistema apmokyta aptikti 5 skirtingas broko rūšis. Sukurta sistema teisingai priima sprendimus esant per dideliu ar per mažam medžiagos kiekiui defektui ir keičia temperatūrą, iki kol ištaisomas brokas. Esant per mažam medžiagos kiekiui, sistema įvertina šį galimą broką ir didina temperatūrą (atpažinimo tikimybė siekia 30 proc.), o esant per dideliu medžiagos kiekiui – mažina temperatūrą (atpažinimo tikimybė siekia 40 proc.). Ištyrus skirtingų spalvų spausdinimo medžiagas, nustatyta, kad ryškiausiai matomos objekto savybės ir brokas naudojant žalios spalvos apšvietimą.

Literatūros sąrašas

1. NGUYEN, D., NGUYEN, C., DUONG-BA, T., NGUYEN, H., NGUYEN, A., TRAN, T. Joint network coding and machine learning for error-prone wireless broadcast. *2017 IEEE 7th Annual Computing and Communication Workshop and Conference (CCWC)* [interaktyvus]. 2017 [žiūrėta 2021-03-22]. Prieiga per: mdpi.com/2073-4360/11/7/1094.
2. KUN, K. Reconstruction and development of a 3D printer using FDM technology. *Procedia Engineering* [interaktyvus]. 2016, 149, 203–211 [žiūrėta 2021-03-22]. Prieiga per: <https://www.sciencedirect.com/science/article/pii/S1877705816311651>.
3. WANG, J., GOYANES, A., GAISFORD, S., ABDUL, W. Stereolithographic (SLA) 3D printing of oral modified-release dosage forms [interaktyvus]. 2018 [žiūrėta 2021-03-22]. Prieiga per: <https://discovery.ucl.ac.uk/id/eprint/1477047/1/SLA%20IJP%20text%20only.pdf>.
4. SANTECCHIA, E., SPIGARELLI, S., CABIBBO, M. Material reuse in laser powder bed fusion: Side effects of the laser – Metal powder interaction. *Metals* [interaktyvus]. 2020, 10, 341 [žiūrėta 2021-03-22]. Prieiga per: <https://www.mdpi.com/2075-4701/10/3/341>.
5. GUNAYDIN, K., TÜRKMEN, H. Common FDM 3D printing defects. *International Congress on 3D Printing (Additive Manufacturing) Technologies and Digital Industry* [interaktyvus]. 2018 [žiūrėta 2021-03-22]. Prieiga per: https://www.researchgate.net/profile/Kadir-Gunaydin/publication/326146283_Common_FDM_3D_Printing_Defects/links/5b4127a1458515f71cb118f7/Common-FDM-3D-Printing-Defects.pdf.
6. SONG, R., TELENKO, C. Material waste of commercial FDM printers under realistic conditions. *An Additive Manufacturing Conference* [interaktyvus]. 2016 [žiūrėta 2021-03-22]. Prieiga per: <http://utw10945.utweb.utexas.edu/sites/default/files/2016/099-Song.pdf>.
7. MATSUZAKI, R., UEDA, M., NAMIKI, M., *et al.* Three-dimensional printing of continuous-fiber composites by in-nozzle impregnation. *Scientific Reports* [interaktyvus]. 2016, 6, 23058 [žiūrėta 2021-03-22]. Prieiga per: <https://www.nature.com/articles/srep23058>.
8. YAZHOU, T., MD IMRAN, H., SHENGMIN, G., XIALI, H. A survey on limitation, security and privacy issues on additive manufacturing. *Cryptography and Security* [interaktyvus]. 2021 [žiūrėta 2021-03-22]. Prieiga per: https://www.researchgate.net/publication/350005263_A_Survey_on_Limitation_Security_and_Privacy_Issues_on_Additive_Manufacturing#pf9.
9. STRAUB, J. Initial work on the characterization of additive manufacturing (3D printing) using software image analysis. *Machines* [interaktyvus]. 2015, 3, 55–71 [žiūrėta 2021-03-22]. Prieiga per: <https://doi.org/10.3390/machines3020055> <https://www.mdpi.com/2075-1702/3/2/55/htm>.
10. ZHANG, S. U. Degradation classification of 3D printing thermoplastics using fourier transform infrared spectroscopy and artificial neural networks. *Applied Sciences* [interaktyvus]. 2018, 8, 1224 [žiūrėta 2021-03-22]. Prieiga per: <https://doi.org/10.3390/app8081224>.
11. DONGARE, A. D., KHARDE, R. R., KACHARE, A. D. Introduction to artificial neural network. *International Journal of Engineering and Innovative Technology (IJEIT)* [interaktyvus]. 2012 [žiūrėta 2021-03-22]. Prieiga per: <https://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.1082.1323&rep=rep1&type=pdf>.
12. O'SHEA, K., NASH, R. *An Introduction to Convolutional Neural Networks* [interaktyvus]. 2015 [žiūrėta 2021-03-22]. Prieiga per: <https://arxiv.org/pdf/1511.08458.pdf>.

13. *IJARAI. The Science and Information Organization* [interaktyvus]. 2013 [žiūrėta 2021-03-22]. Prieiga per: <https://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.278.5274&rep=rep1&type=pdf#page=41>.
14. DONG, N., CANH, N., TUAN, T. Joint network coding and machine learning for error-prone wireless broadcast. *2017 IEEE 7th Annual Computing and Communication Workshop and Conference, CCWC 2017* (2017) [interaktyvus]. 2017 [žiūrėta 2021-03-22]. Prieiga per: https://www.researchgate.net/publication/314202159_Joint_network_coding_and_machine_learning_for_error-prone_wireless_broadcast
15. BAJRAMI, G., DERAWI, M. O., BOURS, P. Towards an automatic gait recognition system using activity recognition (wearable based). *2011 Third International Workshop on Security and Communication Networks (IWSCN)* [interaktyvus]. 2011 [žiūrėta 2021-03-22]. Prieiga per: https://www.researchgate.net/publication/265985834_Towards_an_Automatic_Gait_Recognition_System_using_Activity_Recognition_Wearable_Based.
16. ALBAWI, S. MOHAMMED, T. A., AL-ZAWI, S. Understanding of a convolutional neural network. *2017 International Conference on Engineering and Technology (ICET)* [interaktyvus]. 2017 [žiūrėta 2021-03-22]. Prieiga per: <https://ieeexplore.ieee.org/abstract/document/8308186>.
17. MAHMOOD, M. A., VISAN, A. I., RISTOSCU, C., MIHAILESCU, I. N. Artificial neural network algorithms for 3D printing. *Materials* [interaktyvus]. 2021, 14, 163 [žiūrėta 2021-03-22]. Prieiga per: <https://doi.org/10.3390/ma14010163>.
18. AL-SAFFAR, A. A. M., TAO, H., TALAB, M. A. Review of deep convolution neural network in image classification. *International Conference on Radar, Antenna, Microwave, Electronics and Telecommunications* [interaktyvus]. 2017 [žiūrėta 2021-03-22]. Prieiga per: <https://core.ac.uk/download/pdf/275672004.pdf>.
19. ZHANG, Y., HONG, G. S., YE, D., ZHU, K., FUH, J. Y. H. Extraction and evaluation of melt pool, plume and spatter information for powder-bed fusion AM process monitoring. *Materials & Design* [interaktyvus]. 2018, 156, 458–469 [žiūrėta 2021-03-22]. Prieiga per: <https://www.sciencedirect.com/science/article/abs/pii/S026412751830532X?via%3Dihub>.
20. URBAN, S., LEITLOFF, J., HINZ, S. Improved wide-angle, fisheye and omnidirectional camera calibration. *Journal of IATEX Templates* [interaktyvus]. 2015 [žiūrėta 2021-03-22]. Prieiga per: https://www.ipf.kit.edu/downloads/Preprint_ImprovedWideAngleFisheyeAndOmnidirectionalCameraCalibration.pdf.
21. JINLIANG, L., BU, F. L. Improved RANSAC features image-matching method based on SURF. *The Journal of Engineering* [interaktyvus]. 2019, 23, 9118–9122 [žiūrėta 2021-03-22]. Prieiga per: <https://ietresearch.onlinelibrary.wiley.com/doi/pdfdirect/10.1049/joe.2018.9198>.
22. KEDZIERSKI, M., FRYSKOWSKA, A. Precise method of fisheye lens calibration. *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences* [interaktyvus]. 2008 [žiūrėta 2021-03-22]. Prieiga per: https://www.isprs.org/proceedings/xxxvii/congress/5_pdf/134.pdf.
23. *The Spaghetti Detective* [interaktyvus]. 2021 [žiūrėta 2021-12-10]. Prieiga per: <https://www.thespaghettidetector.com/#gallery>.
24. *Raspberry Pi* [interaktyvus]. 2021 [žiūrėta 2021-11-10]. Prieiga per: <https://www.raspberrypi.com/products/raspberry-pi-4-model-b/specifications/>.

25. *Logitech* [interaktyvus]. 2021 [žiūrėta 2021-11-10]. Prieiga per: <https://www.logitech.com/en-us/products/webcams/c920e-business-webcam.960-001401.html#specs>.
26. *MakerBot Thingiverse* [interaktyvus]. 2021 [žiūrėta 2022-03-17]. Prieiga per: <https://www.thingiverse.com/thing:3114849>.
27. *Raspberry Pi* [interaktyvus]. 2021 [žiūrėta 2021-11-10]. Prieiga per: <https://www.raspberrypi.com/documentation/accessories/camera.html>.
28. *Sb components* [interaktyvus]. 2021 [žiūrėta 2021-11-10]. Prieiga per: <https://shop.sb-components.co.uk/blogs/posts/raspbian-buster-the-new-default-operating-system-for-raspberry-pi>.
29. ABADI, M. Tensorflow: A system for large-scale machine learning. *12th USENIX Symposium on Operating Systems Design and Implementation (OSDI 16)* [interaktyvus]. 2016 [žiūrėta 2021-12-10]. Prieiga per: <https://www.usenix.org/system/files/conference/osdi16/osdi16-abadi.pdf>.