

Penghitung Jumlah Tumpukan dan Penentu Tipe Koin Berdasarkan Intensitas Cahaya Baris

Darmawan Utomo

Program Studi Teknik Komputer,
Fakultas Teknik Elektronika dan Komputer,
Universitas Kristen Satya Wacana, Salatiga
darmawan.utomo@uksw.edu

Abstrak

Alat ukur yang dapat menghitung jumlah tumpukan barang secara otomatis dan cepat sangat dibutuhkan industri dan bisnis. Teknik menghitung jumlah koin, pada umumnya menggunakan mesin yang secara fisik dapat merusak, dan menghasilkan polusi, dan jika mengandung virus dan bakteri dapat menyebarkan penyakit. Untuk itu pada makalah ini diusulkan sebuah penghitung jumlah dan tipe tumpukan koin dengan menggunakan intensitas cahaya baris sebagai pendeteksi objek koin. Tipe atau nominal koin ditentukan dari kombinasi fungsi kepadatan probabilitas jarak antar puncak dan iluminasinya. Jumlah koin dapat dihitung berdasarkan koin-koin yang telah dikenali. Pada 70 sampel citra uang koin 1, 5, 10, dan 50 NTD, diperoleh akurasi pendeteksian koin mencapai 98,98%, dan berhasil menentukan koin dengan benar sebesar 88,6%.

Kata kunci: *stack counter*, koin, intensitas cahaya, pengidentifikasi objek, pengenalan objek

Abstract

Measuring tools that can calculate the number of piles of goods automatically and quickly are needed by industry and businesses. The technique of counting the number of coins generally uses a machine that can physically damage, and produce pollution, and if it contains viruses and bacteria it can spread disease. For this reason, this paper proposes a counter for the number and type of piles of coins using line light intensity as an object detector. The object type is determined by the pixel width of the coin illumination. The number of coins can be calculated based on predefined objects. On 70 samples of 1, 5, 10, and 50 NTD coin images, the accuracy reached 98.98%, and managed to determine the coin correctly by 88.6%.

Keywords: *stack counter*, coin, light intensity, object identification, object recognition

1. Pendahuluan

Industri, peneliti, dan bisnis sangat membutuhkan alat yang dapat menghitung jumlah serta tipe objek dengan cepat dan tepat. Menghitung jumlah objek secara manual pada dunia industri dan riset biologi sel misalnya, sangat memerlukan waktu yang banyak jika dikerjakan secara manual. Pekerjaan ini juga mudah salah oleh karena sifat keletihan pada manusia. Akibat dari kesalahan menghitung jumlah objek ini dapat menimbulkan kesalahan seperti jumlah produksi yang tidak konsisten dalam pembukuan, hasil analisa yang tidak akurat, dan kerugian lainnya.

Teknik-teknik penghitungan ini pada umumnya dimulai dari pendeteksi objek yang akurat. Setelah itu diikuti dengan mendeteksi jumlah dari objek yang dimaksud. Untuk mendeteksi objek secara akurat teknik yang umum digunakan adalah secara manual, transisi logika sensor, dan dengan menggunakan pemrosesan citra. Pada teknik manual cocok untuk objek yang sangat kecil, dan jarang terjadi. Teknik transisi logika sensor banyak diterapkan dengan asumsi adanya transisi logika sensor dari setiap objek. Pada umumnya hanya digunakan menghitung jumlah objek dan tidak peduli dengan tipe dari objeknya. Pada teknik transisi logika, dua objek yang berbeda akan dianggap sama karena single sensor kurang dapat membedakan objek-objek yang berbeda secara luas.

Teknik yang lebih fleksibel adalah dengan menggunakan pemrosesan citra. Ukuran citra tinggi kali lebar seakan-akan diambil dari sejumlah sensor dengan ukuran yang sama. Oleh karena itu, sangat dimungkinkan untuk mendapatkan informasi yang lebih luas dari pada teknik transisi logika. Berbagai macam teknik untuk mendeteksi dan mengidentifikasi objek pada sebuah citra. Teknik pencocokan pola biasa dipergunakan untuk mendeteksi objek berdasarkan pola yang seragam dengan *template*-nya. Teknik seperti scale invariant feature transform (SIFT) [1] dan speeded up robust features (SURF) [2] adalah contoh mengidentifikasi objek secara pemrosesan citra dengan mencari fitur-fitur yang dominan. Teknik yang lebih modern adalah pendeteksi citra adalah dengan menggunakan *machine learning* (ML) dan deep neural networks (DNN) [3-4]. Teknik dengan ML dan DNN sangat tergantung dari koleksi data dan proses pelatihannya. Salah satu teknik untuk mendeteksi objek adalah dengan menggunakan You Look Only Once (YOLO) [5]. YOLO membutuhkan banyak objek untuk pemrosesan training serta memerlukan banyak parameter. Jumlah parameter yang terlalu banyak akan membutuhkan komputasi yang makin besar. Oleh karena itu pada alat ukur yang bersifat bergerak, tanpa internet, dan *standalone*, penggunaan ML/DNN masih sangat jarang. Penelitian yang berkaitan dengan menghitung objek benda banyak dilakukan pada mobil dan pejalan kaki [6-9].

Salah satu contoh produk yang secara komersial telah diterapkan sebagai penghitung objek adalah produk dari Countthings [9-10] dan review produknya dapat dilihat pada [11]. Countthings menggunakan template yang sudah dilatih berdasarkan produk-produk yang ingin dihitung. Beberapa pengguna dari produk ini menganggap hasil dari deteksi dan countingnya kurang akurat. Kelemahan kedua adalah diperlukan template yang cukup memadai untuk dilatih sehingga akurasi bisa ditingkatkan. Kelemahan ketiga adalah harganya yang berkisar 100 USD/bulan perperangkat. Kelemahan keempat adalah kebutuhan memori yang cukup besar untuk menyimpan template citra objek yang diinginkan. Produk ini kurang cocok untuk diterapkan pada alat hitung koin, kertas, uang, atau tumpukan benda lainnya yang pada umumnya tidak berharga mahal dan tidak membutuhkan langganan.

Oleh karena itu, pada makalah ini diusulkan sebuah teknik sederhana untuk mengukur jumlah objek yang dalam hal ini koin, serta menentukan tipe dari koin berdasarkan iluminasinya. Sebuah purwarupa telah dibangun dan teknik evaluasi dipergunakan untuk menghitung jumlah nominal dan tipe koin adalah akurasi.

2. *Digital Color Image (DCI)*

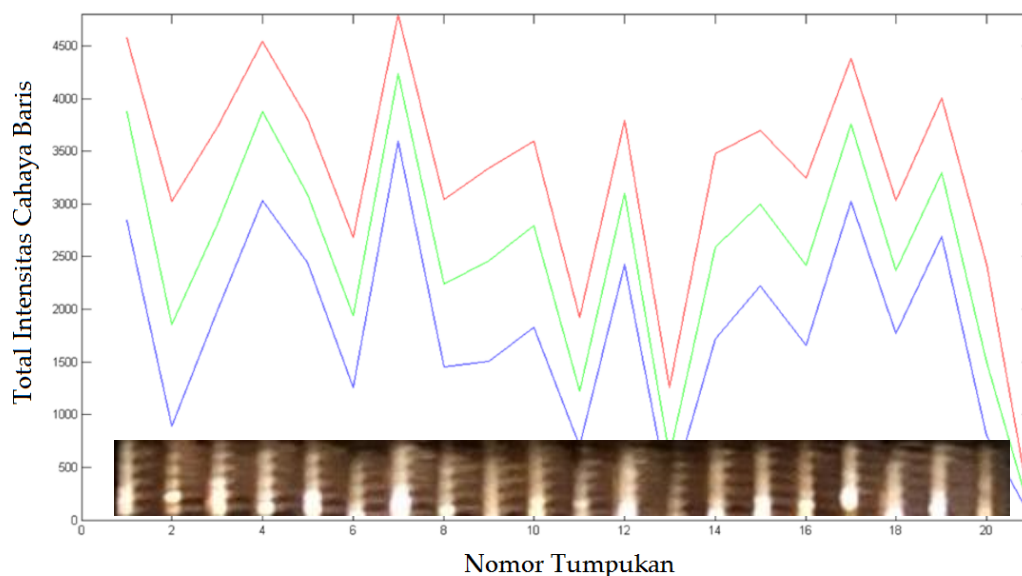
DCI merupakan citra dengan representasi tiga warna (R, G, B) untuk setiap pikselnya [12-13]. Untuk setiap komponen warna memiliki nilai range antara 0 – 255 (1 Byte). Oleh karena itu ada 256^3 variasi warna. Setiap komponen warna dapat direpresentasikan sebagai matriks 2-D pada sebuah citra. Nilai 0 dan 255 mewakili tingkat gelap dan terang maksimum. Data-data nilai dari RGB dari sebuah citra dapat dipergunakan untuk mendapatkan sebuah informasi yang disebut intensitas cahaya baris (*row light intensity*). Dimisalkan terdapat sebuah citra yang menampilkan tumpukan koin, intensitas dari setiap baris ditambahkan sehingga didapatkan intensitas cahaya baris total.

Tabel 1. Ilustrasi intensitas cahaya baris dan representasi gambar sederhana

No Baris	Intensitas Cahaya Baris	←	Nilai Piksel Kolom-1	Nilai Piksel Kolom-2	Nilai Piksel Kolom-3	Nilai Piksel Kolom-4	Nilai Piksel Kolom-5
1	300	←	0	100	100	100	0
2	500	←	100	100	100	100	100
3	500	←	100	100	100	100	100
4	100	←	0	0	100	0	0
5	500	←	100	100	100	100	100
6	500	←	100	100	100	100	100
7	300	←	0	100	100	100	0
8	0	←	0	0	0	0	0
9	400	←	0	100	100	100	100

Tabel 1 memberikan ilustrasi penghitungan intensitas cahaya baris dari sebuah citra dengan ukuran 9×5 . Area abu-abu adalah contoh gambar dengan kumpulan piksel. Kolom intensitas cahaya baris didapatkan dengan menjumlahkan semua piksel dalam satu baris. Dari kolom intensitas cahaya baris, terdapat beberapa nilai di atas nol. Satu nilai puncak yang di atas ambang (yaitu P) dipilih terlebih dahulu. Jika diasumsikan ambang batasnya adalah 200, maka jalur 1,2,3,5,6,7, dan 9 dipilih sebagai puncak-puncaknya (*peaks*) (P1,P2, P3, P5, P6, P7, P9). Berdasarkan percobaan, ketebalan koin dapat dihitung (yaitu 3 baris). Kemudian, dicari puncak maksimum pertama di antara garis 1-3. Dalam hal ini P1 adalah jalur 2. Puncak temuan berikutnya yang akan ditemukan adalah mulai dari jalur-4 hingga jalur-6, kemudian jalur-7 hingga jalur 9. Puncak temuan dalam hal ini adalah pada baris-2 (P1), Baris-5 (P2), dan Baris-9 (P3). Baris-baris ini adalah kandidat koin.

Jenis koin dapat dideteksi menggunakan jarak antara puncak dan warna koin. Jarak antar koin dapat dihitung dengan mengurangkan P2 dengan P1, dan P3 dengan P2, dan seterusnya. Hasil yang ditemukan tidak ideal dan berfluktuasi sehingga jarak rata-rata digunakan sebagai acuan.



Gambar 1. Komponen tiga warna dasar dari sebuah citra terotasi dari tumpukan koin yang telah dilakukan pemrosesan awal. Garis-garis intensitas naik-turun vertikal adalah gerigi-gerigi yang terdapat pada koin. Total intensitasnya berdasarkan warna dasar merah (R), hijau (G), dan biru.

Ada tiga komponen warna dalam sebuah gambar berwarna. Gambar 1 menunjukkan contoh dari sebuah citra dengan 20 koin 1 NTD. Sumbu x menunjukkan jumlah puncak yang terdeteksi. Puncak-puncak ini menunjukkan bahwa nilai intensitas komponen merah selalu lebih tinggi daripada hijau dan biru. Nilai puncak berfluktuasi, tetapi ada relasi yang jelas dari ketiga warna ini. Sehingga fitur yang digunakan bisa memilih satu di antaranya. Pada purwarupa sistem yang dirancang selalu mendapatkan background gelap sehingga tidak diperlukan penghilangan background seperti pada teknik-teknik di makalah [14-15].

3. Metode Penelitian

Ada lima langkah yang akan digunakan dalam percobaan ini untuk mencari jumlah koin dan juga untuk mengetahui apakah tumpukan koin adalah 1 NT, 5 NT, 10 NT, atau 50 NT. Langkah-langkah ini adalah sebagai berikut:

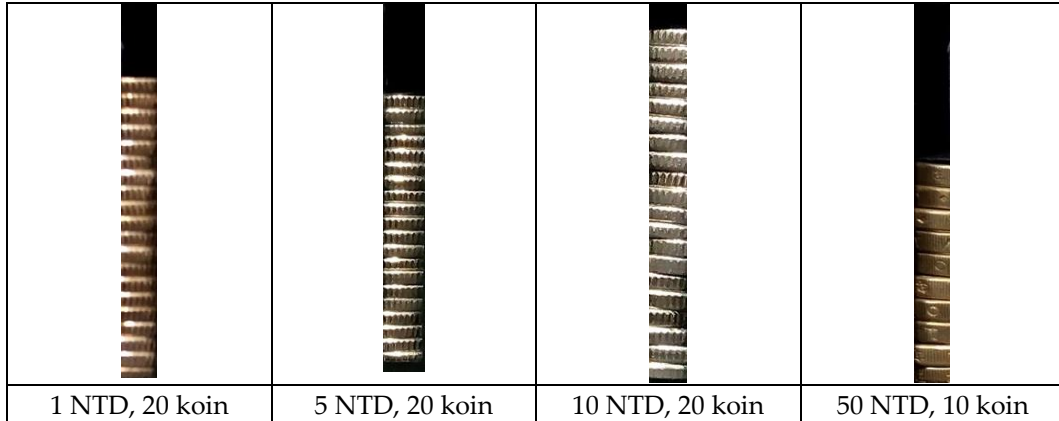
1. Pengumpulan data
2. Pra-pemrosesan gambar dan menemukan intensitas cahaya baris
3. Menemukan ambang batas terbaik
4. Menghitung puncak
5. Menentukan jenis koin berdasarkan jarak dan warna

Data citra yang dikumpulkan sebanyak 70 file dan isi setiap file adalah citra pemotongan tumpukan koin dari 1 hingga 20 koin jenis 1 NTD, 5 NTD, dan 10 NTD. Ada juga 1 hingga 10 koin dalam jenis 50 NTD.

3.1. Pengumpulan Data

Data percobaan dibuat dengan mengambil gambar melalui kamera 3 MP. Terdapat 70 citra dan isi setiap citra adalah gambar potongan tumpukan koin dari 1 hingga 20 koin

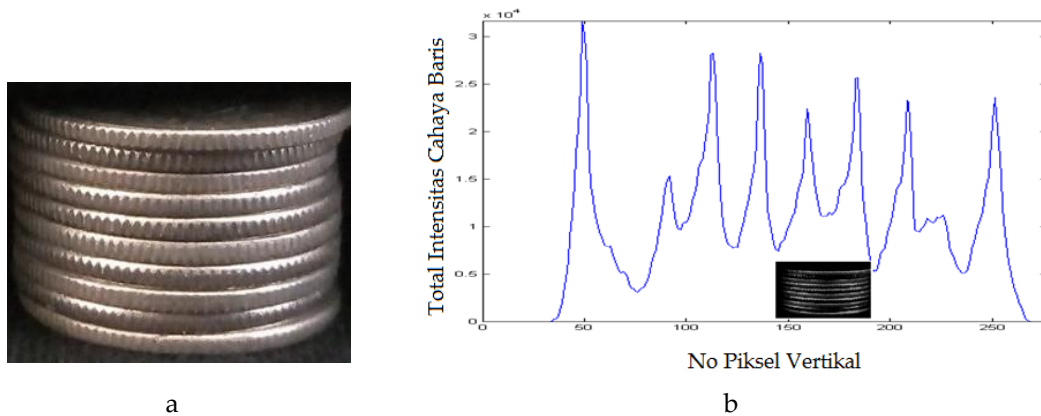
jenis 1 NTD, 5 NTD, dan 10 NTD. Ada juga 1 hingga 10 koin dalam jenis 50 NTD. Gambar 2 menampilkan beberapa file citra yang dijadikan dataset.



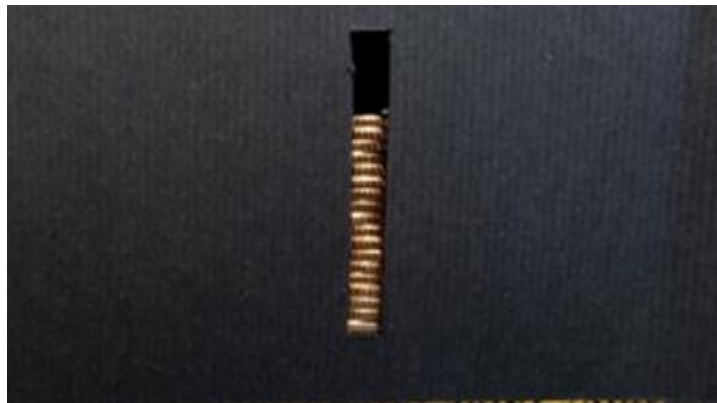
Gambar 2. Beberapa file citra yang menampilkan sampel tumpukan koin.

3.2. Pra-pemrosesan Citra dan Teknik Menemukan Intensitas Cahaya Baris

Citra dari Gambar 3.a menampilkan tumpukan koin 10 NTD sebanyak 10 koin. Jika citra ini diubah ke intensitas cahaya baris akan menghasilkan grafik seperti pada Gambar 3.b. Jika puncak dari grafik ini ditotal jumlahnya kurang dari 10. Oleh karena itu dilakukan modifikasi cara pengambilan citra dengan membatasi kelengkungan dari koin. Gambar 4 menampilkan purwarupa modifikasi pengambilan citra yang membatasi kelengkungan koin ini.

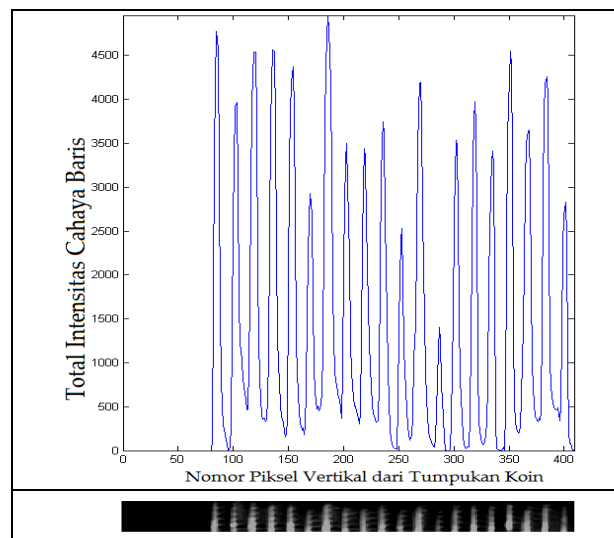


Gambar 3. (a) Lengkungan tumpukan 10 NTD, (b) distorsi yang dihasilkan



Gambar 4. Modifikasi untuk pengambilan citra koin yang mengurangi kelengkungan koin.

Gambar 5 menampilkan hasil dari pra-pemrosesan citra dari citra asli yang telah dibatasi kelengkungannya. Citra aslinya terletak di bawah grafik intensitas. Pada citra asli, terdapat warna hitam yang merupakan lokasi kosong. Sebelum diproses selanjutnya, bagian ini perlu dihilangkan dari data. Data-data lainnya disimpan untuk pemrosesan selanjutnya.



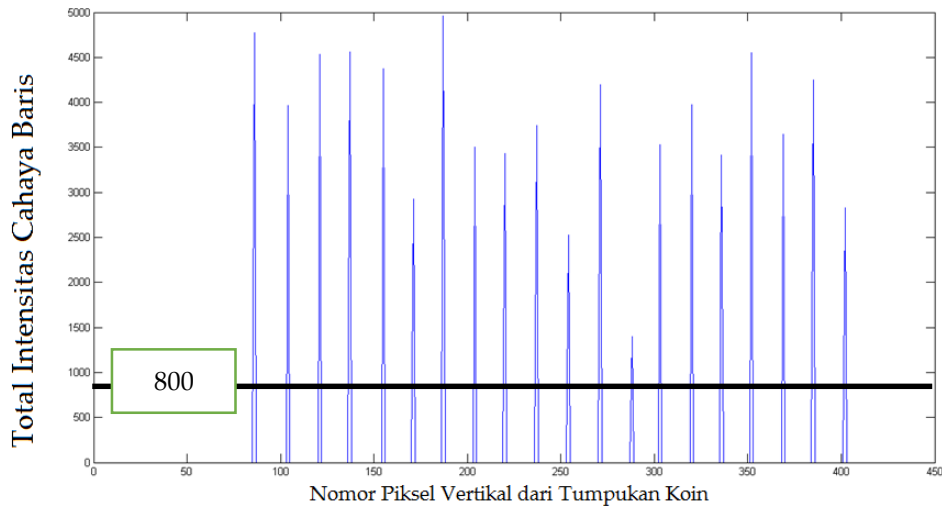
Gambar 5. Hasil intensitas cahaya baris dengan citra koin yang ditampilkan 90° searah jarum jam.

3.3. Mencari Ambang Batas (*Threshold*) Terbaik

Gambar 6 menunjukkan intensitas baris setelah dilakukan proses pencarian titik puncak dan membatasi lebar dari setiap puncak. Beberapa puncak terlihat jelas dan jika dihitung, total puncaknya adalah 20 sama dengan jumlah koin. Namun, tinggi koin tidak sama sehingga kita perlu menemukan ambang batas terbaik untuk semua file sampel.

Untuk mendapatkan ambang yang sesuai, puncak dibandingkan dengan lingkungannya dan beberapa nilai ambang saat proses pencarian ambang yang tepat. Jika puncaknya benar-benar puncak di antara beberapa baris berikutnya dan di atas ambang batas, maka puncak itu dipilih sebagai kandidat koin. Tingkat ambang batas dapat diubah melalui parameter *threshold* dan peningkatan jarak antar puncak. Dalam hal ini, ambang

yang dipilih adalah 800 berdasarkan percobaan untuk kondisi purwarupa ini. Pada total intensitas baris 800, kedupuluh koin ini dapat dikenali semua.



Gambar 6. Puncak setelah menghilangkan puncak non-kandidat.

3.4. Menghitung Jumlah Puncak

Puncak yang terdeteksi harus berada dalam jarak tertentu dengan puncak lainnya. Jarak terhadap puncak lainnya dapat digunakan untuk menentukan tipe koin. Pada percobaan ini diasumsikan tumpukan koin tersusun secara meningkat dari nilai yang kecil ke besar.

4. Hasil dan Pembahasan

Proses mengenali dan mengidentifikasi koin telah dijelaskan pada bagian ke-3. Pada bagian ini dijelaskan pengujian penghitung jumlah koin dan proses menentukan jumlah koin.

4.1. Penghitung Jumlah Koin

Pada percobaan ini bertujuan mencari akurasi jumlah koin. Terdapat 70 file citra yang berisi 1 NTD hingga 50 NTD. Dari total 685 koin yang tersebar di 70 file, terdeteksi 678 koin sehingga akurasinya adalah 98,98% atau meleset 7 koin. Kesalahan ini terdeteksi oleh karena pencahayaan yang kurang sehingga tidak mampu mencapai ambang batas. Tabel 2 merepresentasikan hasil percobaan ini. Hasil ini cukup menjanjikan karena sampel data diperoleh dari kotak purwarupa yang terbuat dari kertas hitam, hanya dengan pencahayaan ruangan tanpa modifikasi, dan menggunakan kamera dengan 3M pixels. Hasil yang lebih akurat dapat dengan mudah dicapai dengan menggunakan peralatan yang lebih standar dengan lingkungan dan pencahayaan yang lebih terkendali.

Tabel 2. Hasil penghitungan jumlah koin

Total Koin	Salah (koin)	Benar (koin)	Akurasi
685	7	678	98,98%

4.2. Menentukan Tipe Koin

Tabel 3. Menentukan jenis uang logam dalam empat percobaan.

Tipe Percobaan	Prob (R-G) * Prob (G-B)	Prob(DISTANCE)	Prob(DISTANCE) * Prob (G-B)	Mean + Pengaturan Manual
Akurasi (%)	62,9	82,9	88,6	87,1

Percobaan kedua bertujuan untuk mendeteksi tipe dari koin dan dibagi menjadi empat macam percobaan. Eksperimen dilakukan berdasarkan probabilitas dalam warna, jarak, warna dan jarak, dan rata-rata dengan penyesuaian jarak manual. Tabel 3 menunjukkan hasil percobaan.

Dalam percobaan ini istilah warna dasar dalam hal ini adalah intensitas warna merah (R), hijau (G), dan biru (B). Probabilitas yang dimaksudkan di sini adalah mencari nilai dari fungsi kepadatan probabilitas (pdf) dari distribusi normal standar. Nilai R-G adalah selisih intensitas merah terhadap hijau. Probabilitas warna saja tidak dapat mencapai hasil yang baik karena perpotongan antara jenis koin hanya memberikan akurasi 62,9%. Tabel 4 menunjukkan hasil nilai minimum dan maksimum R-G dan R-B untuk beberapa sampel uang logam.

Tabel 4. Area R-G dan R-B

R-G \ Koin (NTD)	1	50	5	10
Min	564	201	130	64
Max	1166	773	344	300

R-B \ Koin (NTD)	1	50	5	10
Min	1000	550	550	400
Max	2000	1900	1124	700

Probabilitas jarak (Distance) menunjukkan hasil yang lebih baik tetapi masih kurang dibandingkan dengan metode Mean dan Pengaturan Manual. Mean dan Pengaturan Manual memberikan hasil salah deteksi sebanyak tujuh koin di antara 70 koin atau memberikan akurasi sebanyak 87,1% seperti yang diperlihatkan pada Tabel 5. Kesalahan ini terjadi oleh karena total intensitas yang kurang. Tampak bahwa nilai koin positif salah selalu lebih kecil dari nilai aktualnya.

Hasil yang terbaik dicapai dengan menggunakan $Prob(DISTANCE) * Prob(G-B)$ yang menghasilkan akurasi sebesar 88,6%. $Prob(G-B)$ dipilih karena memberikan hasil yang lebih baik dari pada $Prob(R-G)$ atau $Prob(R-G) * Prob(G-B)$.

Tabel 5. Hasil dari mean dan penyesuaian manual

Data Index No. (1-70)	Tipe	Salah Mendeteksi (Tipe)
1	1	0
21	5	0
22	5	1
23	5	1
29	5	1
30	5	1
31	5	1
41	10	0
61	50	0
Total		9
Akurasi (%)		87,1

5. Kesimpulan dan Saran

Sebuah metode untuk menentukan nilai dan tipe dari koin dengan teknik intensitas cahaya baris telah dijabarkan. Nilai dari tumpukan koin dapat dideteksi dengan akurasi 98,9%. Sedangkan untuk mendeteksi jenis koin, hasil terbaik dicapai dengan menggabungkan probabilitas jarak dan probabilitas hijau-biru yang menghasilkan akurasi 88,6%. Metode sederhana ini dapat dikembangkan lebih lanjut untuk objek benda yang spesifik dan dengan lingkungan yang terkontrol pencahayaannya. Untuk penelitian selanjutnya perlu diteliti algoritma-algoritma baik dari sisi pemrosesan citra, ML, dan DNN yang dikhususkan untuk diterapkan pada sistem *embedded*.

6. Ucapan Terima Kasih

Penulis mengucapkan terima kasih kepada Universitas Kristen Satya Wacana yang telah mendanai penelitian ini melalui hibah No. 332/Pen./Rek./9/2022.

Daftar Pustaka

- [1] Z. Dai-Xian, "SIFT algorithm analysis and optimization," 2010 International Conference on Image Analysis and Signal Processing, 2010, pp. 415-419, doi: 10.1109/IASP.2010.5476084.
- [2] H. Bay, T. Tuytelaars, L.V. Gool, "SURF: Speeded Up Robust Features," Computer Vision -- ECCV 2006, Springer, 2006, pp. 404-417.
- [3] T. Stahl, S. L. Pintea and J. C. van Gemert, "Divide and Count: Generic Object Counting by Image Divisions," *IEEE Transactions on Image Processing*, vol. 28, no. 2, pp. 1035-1044, Feb. 2019.
- [4] J.P.A. Hoekendijk, B. Kellenberger, G. Aarts, et al. "Counting using deep learning regression gives value to ecological surveys." *Sci Rep* 11, 23209 (2021).
- [5] R. Joseph, S. Divvala, R. Girshick, A. Farhadi, "You Only Look Once: Unified, Real-Time Object Detection," arXiv, 2016. <https://arxiv.org/abs/1506.02640> (Diakses terakhir 30 Maret 2022)
- [6] J. Huang et al., "Drone-Based Car Counting via Density Map Learning," 2020 *IEEE International Conference on Visual Communications and Image Processing (VCIP)*, Macau, China, 2020, pp. 239-242.

- [7] Jiajun Lu, Yi Xu and Xiaokang Yang, "Counting pedestrians and cars with an improved virtual gate method," *2010 International Conference on Computer Application and System Modeling (ICCASM 2010)*, Taiyuan, 2010, pp. V4-448-V4-452.
- [8] H. Zhou, L. Wei, M. Fielding, D. Creighton, S. Deshpande and S. Nahavandi, "Car park occupancy analysis using UAV images," *IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, Banff, AB, Canada, 2017, pp. 3261-3265.
- [9] T. Moranduzzo and F. Melgani, "Automatic Car Counting Method for Unmanned Aerial Vehicle Images," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 52, no. 3, pp. 1635-1647, March 2014.
- [10] Countthings, "Apps to Automate Your Counting," Countthings, 2022. <https://countthings.com/> (diakses tanggal 30 Maret 2022).
- [11] Google Play, "User Reviews," Countthings, 2022. <https://play.google.com/store/apps/details?id=com.dyve.countthings&showAllReviews=truehttps://countthings.com/> (diakses tanggal 30 Maret 2022).
- [12] R.C. Gonzalez, R.E. Woods, "Digital Image Processing 4th ED," Pearson, 2018.
- [13] P. Cattin, "Digital Image Fundamentals," University of Basel. [https://miac.unibas.ch/SIP/02-Fundamentals.html#\(1\)](https://miac.unibas.ch/SIP/02-Fundamentals.html#(1)) (Akses terakhir 10 Juni 2022)
- [14] C. Li, S. Ran and L. Lin, "Comparative Research of Dynamic Target Detection Algorithms Based on Static Background," *2021 Photonics & Electromagnetics Research Symposium (PIERS)*, Hangzhou, China, 2021, pp. 2107-2112.
- [15] T. Huynh-The, O. Banos, S. Lee, B. H. Kang, E. -S. Kim and T. Le-Tien, "NIC: A Robust Background Extraction Algorithm for Foreground Detection in Dynamic Scenes," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 27, no. 7, pp. 1478-1490, July 2017.