

Citra Sitentik Untuk Klasifikasi Buah Menggunakan Algoritma SIFT Descriptor, Bag of Features dan Support Vector Machine

Achmad Lukman^{1,*}, Erni Seniwati², Eko Riswanto³

¹ Fakultas informatika, Teknologi Informasi, Telkom University, Bandung, Indonesia

² Fakultas Ilmu Komputer, Sistem Informasi, Universitas Amikom Yogyakarta, Yogyakarta, Indonesia

³ Program Studi, Informatika, STMIK El Rahma, Yogyakarta, Indonesia

Email: ^{1,*}alukman@telkomuniversity.ac.id, ²erni.s@amikom.ac.id, ³riswantoeko@stmikelahma.ac.id

Email Penulis Korespondensi: alukman@telkomuniversity.ac.id

Submitted: 17/11/2024; Accepted: 17/12/2024; Published: 18/12/2024

Abstrak—Mengenali objek spesifik yang ditugaskan kepada computer menggunakan artificial intelligence tentu melewati proses *training* dan *testing* dengan metode mesin *learning*, terbatasnya jumlah *dataset* menjadi hal yang sulit bagi metode *deep learning* untuk melakukan klasifikasi, sehingga untuk mengatasi hal tersebut perlu metode lain diantaranya *Scale Invariant Features Transform* (SIFT) yang merupakan salah satu metode dari pengolahan citra untuk mengekstraksi fitur dari jumlah data yang terbatas dan dikombinasikan dengan salah satu metode didalam mesin *learning*. Untuk mengatasi ketidakmampuan *deep learning* dengan menggunakan *dataset* yang terbatas tersebut, maka di dalam penelitian ini menggunakan kombinasi SIFT dan *bag of features* untuk melakukan ekstraksi fitur dan support vector machine (SVM) untuk melakukan klasifikasi. Pada penelitian ini, bertujuan mengamati efek citra sintetik terhadap kinerja kombinasi algoritma *SIFT descriptor*, *Bag of Features* dan *Support Vector Machine* dalam melakukan klasifikasi citra buah yang nyata. *Dataset* yang dilibatkan adalah citra sintetik berupa gambar 3D yang dibuat menjadi sebuah objek yang utuh, kemudian mengambil view secara acak untuk dijadikan citra yang mewakili objek tersebut sebagai *data training*. Selanjutnya, untuk data testing akan digunakan citra nyata yang diambil pada *link dataset* pada penelitian sebelumnya. Jumlah *dataset* sintetik yang dapat dikumpulkan untuk masing-masing buah adalah 150 citra sehingga totalnya ada 450 citra, sedangkan citra buah nyata komposisinya 148 citra apel, 152 citra pisan, dan 166 citra jeruk sehingga total citra nyata adalah 466 citra. Hasil penelitian ini memperlihatkan bahwa akurasi tertinggi didapatkan 65.45% dengan F1-score mencapai 58.45%.

Kata Kunci: Bag of Features; Citra Sitentik; Klasifikasi Buah; SVM; SIFT

Abstract—Recognizing specific objects assigned to a computer using artificial intelligence of course goes through a training and testing process using machine learning methods, the limited number of datasets makes it difficult for deep learning methods to carry out classification, so to overcome this, other methods are needed, including Scale Invariant Features Transform (SIFT) which is a method of image processing to extract features from a limited amount of data and combined with a method in machine learning. To overcome the inability of deep learning to use limited datasets, this research uses a combination of SIFT and bag of features to extract features and support vector machine (SVM) to carry out classification. In this study, the aim is to observe the effect of synthetic images on the performance of the combination of SIFT descriptor, Bag of Features and Support Vector Machine algorithms in classifying real fruit images. The dataset involved is a synthetic image in the form of a 3D image that is made into a complete object, then taking random views to make an image that represents the object as training data. Furthermore, for testing data, real images taken from the dataset link in previous research will be used. The number of synthetic datasets that can be collected for each fruit is 150 images, so that the total is 450 images, while the real fruit images consist of 148 apple images, 152 banana images, and 166 orange images, so that the total real images are 466 images. The results of this research show that the highest accuracy was 65.45% with an F1-score reaching 58.45%.

Keywords: Bag of Features; Syntethic Image; Fruit Classification; SVM; SIFT

1. PENDAHULUAN

Klasifikasi buah cukup penting pada area perkebunan, apalagi saat panen tiba. Pemilihan dan klasifikasi sangat dibutuhkan para petani buah untuk memilih dan mengelompokkan jenis buah agar bisa dengan cepat melakukan pengemasan untuk dikirim ke pasar atau *supermarket* [1]. Pada saat bersamaan klasifikasi buah juga membantu para petani buah untuk mencari tahu perlakuan tanaman yang berbeda satu sama lain dalam hal mengelola nutrisi tanaman tersebut, sehingga dapat meningkatkan efisiensi dalam mencapai kualitas produk yang tinggi untuk para konsumen [2][3][4]. Buah-buahan adalah salah satu bahan mentah yang sangat diperlukan oleh industri makanan sehingga klasifikasi buah merupakan proses yang sangat penting baik untuk mengawasi kualitas dan manajemen rantai pasokan[3]. Selain itu, untuk manajemen rantai pasokan dan penelusuran produk, klasifikasi buah dapat membantu melacak sumber dan proses produksi dari produk sehingga memastikan bahwa produk tersebut memenuhi standar kualitas dan keamanan pangan, sehingga meningkatkan kepercayaan konsumen[5][6].

Klasifikasi secara otomatis untuk buah dan sayur telah dikembangkan oleh Huynh, dkk.[7] yang dapat meningkatkan efisiensi klasifikasi. Klasifikasi buah dan sayur secara tradisional sering dilakukan oleh para petani, namun cara tersebut membutuhkan waktu yang lama dan kemungkinan kesalahan yang terjadi lumayan besar serta membutuhkan biaya yang tidak sedikit, sehingga diperlukan klasifikasi menggunakan sistem secara otomatis, apalagi untuk pasokan buah yang jenisnya bervariasi dan jumlahnya lebih besar. Klasifikasi otomatis dapat mengurangi kesalahan yang disebabkan faktor manusia diantaranya bisa salah memilih buah karena kelelahan dan kurang fokus, kecepatan memilih buah yang rendah dibandingkan menggunakan mesin, sehingga pilihan menggunakan klasifikasi otomatis dapat meningkatkan konsistensi dan akurasi pemilihan buah[8]. Memanfaatkan teknologi klasifikasi buah

dan sayuran dengan data yang cukup besar dapat diproses secara cepat untuk meningkatkan efisiensi produksi dan efisiensi operasi penjualan[2].

Keterbatasan untuk mendapatkan *dataset* yang di inginkan terutama yang tersebar pada internet menjadi salah satu kendala dalam membuat sistem klasifikasi otomatis. Kendala tersebut dapat diatasi dengan membuat sendiri *dataset*, namun perlu waktu dan biaya yang lumayan besar. Untuk itu, penulis menawarkan klasifikasi buah dengan menggunakan *dataset* sintesis yang dibuat dengan citra sintetis 3 dimensi berupa buah pisang, apel, dan jeruk. Kemudian citra buah tersebut di *capture* setiap sisinya secara *random* untuk mendapatkan kumpulan *dataset* yang akan dijadikan data pelatihan. Untuk data pengujian digunakan citra nyata yang diambil dari beberapa penelitian sebelumnya oleh Muresan dan Oltean[9][10]. Selanjutnya, penulis menawarkan kombinasi algoritma SIFT dan *Bag of feature* untuk melakukan ekstraksi fitur, dan algoritma SVM untuk melakukan klasifikasi.

Pada penelitian ini, kami akan melakukan investigasi efek dari *dataset* sintetis yang dibuat terhadap akurasi jaringan *convolutional neural network* yang akan dilibatkan dalam melakukan klasifikasi buah yang sebenarnya. Kontribusi yang ditawarkan pada penelitian ini adalah pertama, menawarkan teknik pengumpulan *dataset* sintetis menggunakan objek 3 dimensi yang dibuat menyerupai buah pisang, buah jeruk dan buah apel, yang mana masing-masing objek tersebut diambil citranya dari berbagai sisi secara *random* sehingga dapat mengumpulkan 150 citra setiap jenisnya. Kedua, menggunakan algoritma SIFT untuk melakukan ekstraksi fitur dari citra sintetis yang telah dikumpulkan pada langkah pertama, pada proses ini setelah citra diubah menjadi *grayscale* (keabuan) maka citra tersebut akan di ekstraksi dengan beberapa langkah yang akan dijelaskan pada bagian ekstraksi fitur pada makalah ini sehingga hasil dari proses ini mendapatkan SIFT *descriptor*. Ketiga, SIFT deskriptor yang telah dihasilkan kemudian diproses pada *Bag of features*, pada langkah ini semua deskriptor akan dikumpulkan sesuai dengan kemiripan orientasi fiturnya menggunakan algoritma K-means dan disajikan dalam bentuk histogram, pada proses ini juga investigasi dilakukan dengan beberapa skenario variasi jumlah kluster yaitu 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, untuk melihat kinerja algoritma yang paling bagus terhadap proses klasifikasi selanjutnya. Tahap ke empat, tahap ini adalah proses klasifikasi menggunakan algoritma *Support vector machine* (SVM), semua fitur yang telah dihasilkan dari proses *Bag of features* dijadikan data latih untuk proses klasifikasi. Pada tahap akhir ini juga observasi variasi *hyperparameter* SVM jenis kernel diantaranya adalah kernel linier, kernel polinomial, dan kernel sigmoid. Semua jenis kernel ini diterapkan pada masing-masing variasi jumlah kluster *Bag of features*. Untuk melihat hasil kinerja kombinasi algoritma tersebut dalam melakukan klasifikasi, maka digunakan pengukuran metrik diantaranya menggunakan akurasi, *precision*, *recall*, dan *F-1 score*.

Selanjutnya penelitian ini akan menjelaskan alur kerja kombinasi algoritma pada bagian 2, termasuk alur penelitian yang kami gunakan. Pada bagian 3, kami akan menguraikan metode penelitian yang digunakan dan juga termasuk hasil dari eksperimen yang kami jalankan. Terakhir pada bagian 4, kami akan menyimpulkan hasil eksperimen dan kemungkinan pengembangan penelitian selanjutnya.

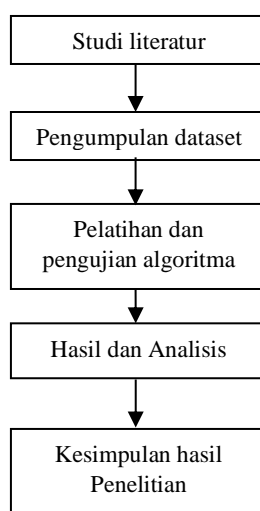
Beberapa penelitian sebelumnya telah melakukan penelitian klasifikasi citra buah diantaranya Huynh, dkk. [7] yang melakukan penelitian tentang klasifikasi sayuran dan buah-buahan secara otomatis, penelitian mereka menggunakan beberapa tahapan, pada tahap pertama mereka menggunakan arsitektur jaringan Densnet121 untuk tujuan mengambil fitur dari *dataset* buah. Pada tahap kedua, mereka melakukan seleksi fitur yang sangat dominan untuk mengenali jenis buah dengan menggunakan beberapa metode antara lain *Adaptive particle – Grey Wolf optimization* (APGWO). Kemudian tahap selanjutnya menggunakan metode *K-Nearest Neighbor* (KNN), *Support Vector Machine* (SVM), *Decision Tree* (DT), *Random Forest* (RF), and *Multilayer Perceptron* (MLP) untuk proses klasifikasi buah. Penelitian yang melakukan klasifikasi buah naga [11] yang menggunakan teknik pengolahan citra digital untuk membuat data augmentasi menggunakan *noise filtering* (*Gaussian filter*), *image quality enhancement* (*image rotation*), *flip the image*, *zoom in*, *zoom out*, dan lain-lain. Penelitian ini menghasilkan *dataset* baru berupa *dataset* buah naga dengan pengambilan citra menggunakan banyak sudut dan kondisi yang berbeda dari *dataset* sejenis seperti suhu, temperatur, dan pencahayaan). Pada penelitian lain nya juga telah dilakukan pengukuran terhadap kemampuan *bags of features*, *convolutional neural network*, *support vector machine*, *multi-features fusion* atau berbasis *deep features* untuk melakukan klasifikasi terhadap beberapa objek yang berbeda dan dilakukan juga pada *dataset* dari bidang keilmuan yang berbeda seperti data cancer atau pun data pertanian yaitu buah-buahan [12]. Penelitian yang dilakukan oleh Behera, dkk.[13], membuat model untuk melakukan klasifikasi terhadap 40 macam jenis buah dari India menggunakan SVM dan ekstraksi fiturnya menggunakan lapisan *fully connected* dari *convolutional neural network*, mereka melakukan perbandingan antara model *fully connected-SVM* dengan transfer *learning* CNN. Hasilnya, model yang mereka tawarkan mendapatkan akurasi 100%. Penelitian selanjutnya yang dilakukan oleh Peng, dkk.[14], melakukan kombinasi algoritma *Convolutional neural network* (CNN) dengan algoritma *Support vector machine* (SVM) untuk meningkatkan kinerja model CNN dalam melakukan klasifikasi Lalat buah dengan latar belakang yang kompleks. Penulis melakukan modifikasi yang mana lapisan konvolusi pada CNN digunakan untuk mengekstraksi fitur lalat buah, selanjutnya fitur tersebut diklasifikasi menggunakan SVM. Hasil penelitian mereka memperlihatkan akurasi mencapai 92.04% pada lalat buah *Bactrocera dorsalis* Hendel, *Bactrocera cucurbitae*, *Bactrocera tau* and *Bactrocera scutellata*. Sejalan dengan penelitian klasifikasi buah menggunakan algoritma SVM, Ge, dkk. [15], menawarkan metode untuk membuat klasifikasi organ pohon apel secara otomatis. Penulis menawarkan metode berbasis warna dan bentuk *Multi-features Fusion* dan model klasifikasi menggunakan SVM. Dalam melakukan panen buah apel dari pohonnya secara efektif, penulis ingin memecahkan masalah daun

pohon apel yang terlalu lebat, hambatan yang parsial dalam melakukan identifikasi citra buah apel matang, dan mengelompokkan citra buah apel langsung pada pohonnya. Untuk mengatasi masalah tersebut, penulis mendesain algoritma yang dirancang untuk pohon apel yang kerdil dan padat pada tahap kematangan awal dan akhir. Vektor fitur yang penulis gunakan adalah 196 dimensi, terdiri dari Vektor fitur 196 dimensi yang terdiri dari *Red Green Blue* (RGB), *Hue Saturation Value* (HSV), *Curvatures*, *Fast Point Feature Histogram* (FPFH), dan *Spin Image* diekstraksi terlebih dahulu. Kemudian SVM berdasarkan fungsi kernel linier dilatih, setelah itu SVM yang dilatih digunakan untuk klasifikasi organ pohon apel. Hasil penelitian mereka memperlihatkan bahwa Nilai *Recall*, *Precision* dan *F1-score* dari metode estimasi yield yang diusulkan masing-masing adalah 93,75%, 96,15% dan 94,93%. Penelitian lain yang dilakukan oleh Arora, dkk.[16], melakukan penelitian diagnosis awal kanker kulit menggunakan *Bag of Feature* dan *support vector machine*. Metode yang diusulkan menunjukkan akurasi 85,7%, sensitivitas 100%, spesifisitas 60% dan waktu pelatihan 0,8507 s dalam mengklasifikasikan luka di kulit.

2. METODOLOGI PENELITIAN

2.1 Tahapan Penelitian

Pada tahap ini, penelitian yang dijalankan terdiri dari beberapa tahapan penting sehingga dapat menjadi sebuah laporan penelitian yang lebih lengkap, urutannya terlihat pada Gambar 1.



Gambar 1. Alur tahapan penelitian

Berdasarkan tahapan penelitian yang terdapat pada Gambar 1, terdapat lima langkah penting yang dijelaskan sebagai berikut.

2.1.1 Studi literatur

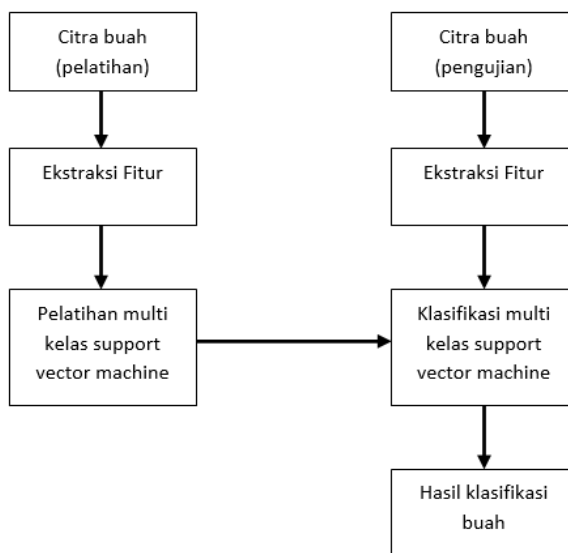
Pada tahap pertama ini, penulis memulai dengan melakukan review terhadap beberapa literature yang berhubungan dengan topik penelitian klasifikasi buah menggunakan beberapa metode visi komputer dan mesin pembelajaran.

2.1.2 Pengumpulan dataset

Tahapan ini, penulis mengumpulkan dataset sintesis untuk keperluan pelatihan mesin pembelajaran dengan membangun obyek tiga dimensi berdasarkan pola gambar buah apel, pisang, dan jeruk menggunakan program java. Kemudian penulis mengambil sisi-sisi dari *view* gambar 3 dimensi tersebut secara random sehingga mendapatkan masing-masing 150 *view* per obyek. Sedangkan untuk dataset pengujian diambil dari penelitian klasifikasi obyek oleh [9] yang terdapat pada link paper tersebut.

2.1.3 Pelatihan dan Pengujian algoritma

Pada tahap ini, kami mendesain algoritma untuk mengekstraksi fitur dari dataset yang telah kami kumpulkan dan mendesain algoritma untuk pelatihan dan pengujian hasil ekstraksi fitur yang didapatkan dari proses sebelumnya. Pada penelitian klasifikasi buah dengan citra sintetik ini melibatkan dua bagian algoritma yaitu algoritma yang digunakan untuk ekstraksi fitur citra dan algoritma yang bertugas untuk klasifikasi citra yang digunakan sesuai dengan urutan penelitian yang dibuat, seperti terlihat pada Gambar 2.



Gambar 2. Bagan alir sistem klasifikasi buah menggunakan data sintetis

Berdasarkan Gambar 2, terdapat pembagian *dataset* yang umum dilakukan yaitu untuk proses pelatihan dan untuk proses pengujian yang proporsinya akan dijelaskan pada bagian eksperimen. Selanjutnya pada bagian ekstraksi fitur menggunakan *scale-Invariant Feature Transform* (SIFT) dan *visual vocabulary* menggunakan K-means dan support vector machine digunakan sebagai metode klasifikasi. Penjelasan metode yang digunakan akan dibahas pada sub bagian selanjutnya. Sedangkan Formula pengukuran metrik yang digunakan pada penelitian ini adalah nilai akurasi yang memanfaatkan metode confusion matrix, yang terlihat pada Gambar 3.

		Nilai aktual	
		Positive	Negative
Nilai Prediksi	Positif	TP (True Positive)	FP (False Positive)
	Negative	TP (True Positive)	FP (False Positive)

Gambar 3. Confusion Matrix.

Berdasarkan Gambar 3, dapat diturunkan formula untuk menghitung akurasi, *precision*, *recall*, dan *F1-score*.

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \tag{1}$$

Persamaan (1), menggambarkan rasio yang memiliki nilai benar (benar positif dan benar negatif) dengan melibatkan keseluruhan data. Selanjutnya menghitung *precision* dan *recall* sebagai berikut.

$$precision = \frac{TP}{TP+FP} \tag{2}$$

$$recall = \frac{TP}{TP+FN} \tag{3}$$

Pada persamaan (2) memperlihatkan perhitungan *precision* yang artinya menghitung data rasio benar-benar positif dibandingkan dengan hasil prediksi positif. Sedangkan persamaan (3) menghitung sensitifitas dari data rasio benar positif dengan data faktual benar positif. Sehingga untuk mengetahui desain algoritma benar dalam melakukan prediksi atau klasifikasi dengan menggabungkan persamaan (2) dan (3) menghasilkan persamaan (4).

$$F1\ score = 2 \times \frac{precision \times recall}{precision+recall} \tag{4}$$

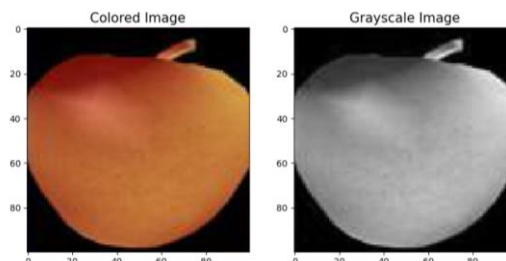
Persamaan (4) memperlihatkan rasio perbandingan rata-rata *precision* dan *recall*.

2.1.3.1 Algoritma Ekstraksi Fitur

Algoritma yang digunakan dalam penelitian ini adalah kombinasi dari *scale-invariant features transform* yang bertugas untuk melakukan ekstraksi fitur dari citra. Selanjutnya, pemilihan fitur yang lebih menonjol dari proses ekstraksi sebelumnya menggunakan *framework Bag of Features* (BoF). Kemudian fitur tersebut masuk ke proses klasifikasi menggunakan *Support Vector Machine* (SVM).

2.1.3.2 Scale-Invariant Features Transform (SIFT)

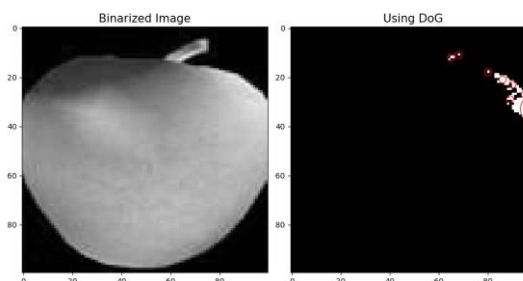
Meskipun ukuran dan bentuk deskriptor yang didapatkan menggunakan metode komputasi geometri lebih efisien untuk klasifikasi, nilai dari semua parameter skala dan resolusi saling bergantung atau dependen [17][18], untuk itulah penulis akan melakukan eksplorasi pengaruh skala dan resolusi citra terhadap kinerja support vector machine. SIFT telah banyak digunakan untuk ekstraksi fitur citra pada proses klasifikasi. Sebelum masuk ke implementasi penggunaan algoritma SIFT, langkah pertama semua citra diubah menjadi *grayscale* seperti Gambar 4.



Gambar 4. Inisialisasi citra dengan kode transformasi dari rgb ke grayscale

Semua citra yang akan diproses menggunakan algoritma SIFT akan diubah menjadi *grayscale* seperti pada Gambar 4. langkah selanjutnya, Ada empat langkah implementasi menggunakan python pada deteksi fitur SIFT yaitu.

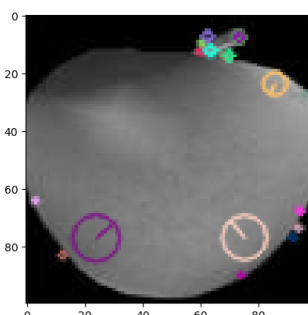
- a. *Scale-space extrema detection*, Mencari blob dengan pendekatan LoG (*Laplacian of Gaussian*) dengan DoG (*Difference of Gaussian*) untuk mendeteksi *interest poin* yang tidak bergantung pada orientasi dan skala seperti pada Gambar 5.



Gambar 5. Deteksi *interest poin* menggunakan DoG (*Different of Gaussian*) [19]

- b. *Keypoint localization*, kemudian lokasi yang terdeteksi pada langkah pertama diperbaiki dengan dua jenis ambang batas, satu untuk kontras (*contrast*) dan satu untuk tepi (*edge*).
- c. *Orientation assignment*, Kemudian, satu atau beberapa orientasi ditetapkan pada setiap *key point* berdasarkan arah gradien citra lokal. Semua operasi selanjutnya dilakukan pada data citra yang telah diubah relatif terhadap orientasi, skala, dan lokasi yang ditetapkan untuk setiap fitur, sehingga memberikan ketidaktergantungan pada transformasi ini.
- d. *Keypoint descriptor*, selanjutnya, di wilayah sekitar setiap *key point*, *descriptor* fitur lokal dihitung. Setiap *descriptor* fitur adalah vektor fitur berdimensi 128 yang secara khusus mengidentifikasi lingkungan sekitar *key point*. Untuk memberikan invariansi orientasi, deskriptor ini didasarkan pada gradien citra lokal, yang diubah sesuai dengan orientasi *key point*.

Contoh implementasi obyek tunggal apel pada langkah b,c dan d yaitu mencari *Keypoint localization*, *Orientation assignment* dan *Keypoint descriptor* dirangkum menggunakan *library* OpenCV yang kode dan hasilnya diperlihatkan pada Gambar 6.



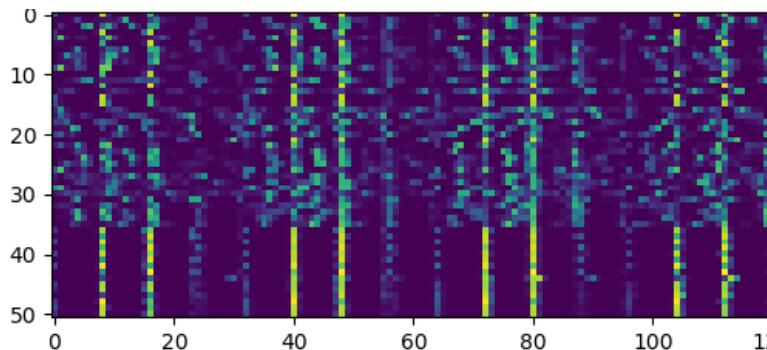
Gambar 6. *Descriptor* fitur dari obyek tunggal

Pada proses ekstraksi fitur menggunakan algoritma SIFT menghasilkan fitur dengan dimensi yang besar sehingga perlu melakukan pengurangan dimensi fitur dengan melakukan pemilihan fitur yang lebih penting.

2.1.3.3 Bag of Features

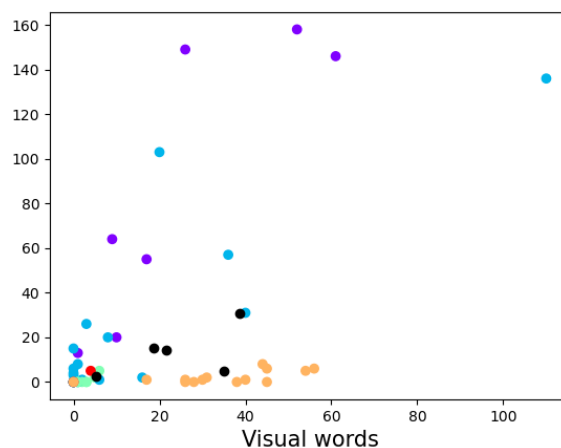
Cara kerja algoritma ini dengan fitur penting sebagai *vocabulary* yang digunakan untuk menjelaskan setiap fitur unik dengan visualisasi histogram. Histogram ini dihasilkan dari kinerja algoritma k-means untuk mengelompokkan fitur secara otomatis. Secara mendasar kinerja dari BOF sebagai berikut.

- a. Mendeteksi descriptor fitur menggunakan SIFT, yang terlihat pada Gambar 7.



Gambar 7. Deteksi *descriptor* fitur dari seluruh *dataset* dengan tiga kelas untuk pelatihan

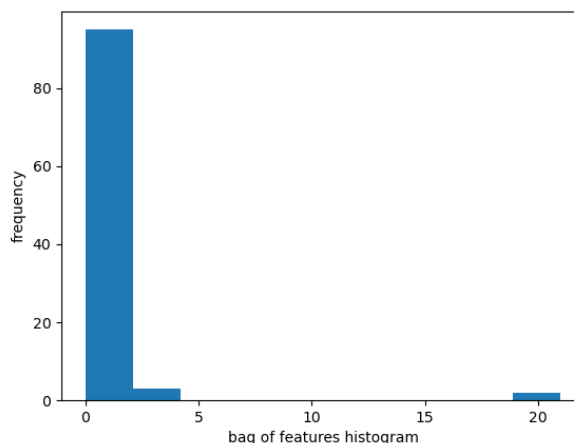
- b. Mengelompokkan descriptor kedalam cluster dengan algoritma kuantisasi vektor menggunakan K-means, terdapat pada Gambar 8.



Gambar 8. Mengelompokkan *descriptor* fitur menggunakan *K-means clustering*

Pada Gambar 8. Memerlihatkan metode K-means mengelompokkan fitur dengan pusat *cluster* berwarna hitam, selainnya adalah *descriptor* fitur. Setiap pusat cluster tersebut bertindak sebagai *visual word* atau *visual of features* yang mewakili fitur unik dari masing-masing citra.

- c. Membangun *bag of feature histogram* seperti diperlihatkan pada Gambar 9.



Gambar 9. Membangun *bag of features histogram*

- Pada Gambar 9. Memperlihatkan histogram dengan jumlah cluster 5, jumlah cluster ini akan diujikan dengan variasi jumlah cluster lainnya.
- d. Proses pelatihan fitur *bag of features* sebagai fitur vektor dan menentukan kategori citra menggunakan algoritma SVM. Proses pelatihan fitur diperlihatkan pada Gambar 10.

```

Input : Training set S (bag of Features) as histogram features; Total number of models used bag of
features = 3 kernels of SVM( linear, polynomial, sigmoid)
Procedure Bag of Features-svm
  Training Process:
    For i ← 1 to 3 do
      Use Histogram features Si number of clusters to
      create the classification models;
    EndFor
  EndProcedure
    
```

Gambar 10. Pseudocode untuk proses pelatihan fitur *Bag of Features*.

Secara efektif, proses ini memaksimalkan akurasi klasifikasi dan meminimalkan proses komputasi [20].

2.1.3.4 Support vector Machine

Algoritma *support vector machine* bekerja sangat baik dalam dimensi data yang lebih besar, untuk mengolah data tersebut terdapat beberapa pilihan fungsi *kernel* untuk melakukan *higher dimensional mapping* sehingga memudahkan proses klasifikasi. Klasifikasi SVM menggunakan prinsip regresi linier untuk kasus linier dan dapat juga menangani kasus non-linier. Pada penelitian ini, fungsi kernel yang digunakan untuk mencari batasan hyper-plane dengan vektor-vektor hasil dari proses *Bag of Feature* yaitu.

$$\text{linear} : K(x_i, x_j) = x_i^T x_j \tag{5}$$

$$\text{Polynomial} : K(x_i, x_j) = (\gamma \cdot x_i^T x_j + r)^d, \gamma > 0 \tag{6}$$

$$\text{Sigmoid} : K(x_i, x_j) = \tanh(\gamma \cdot x_i^T x_j + r) \tag{7}$$

Dengan fungsi regresi yang berupa formula pada persamaan (4).

$$f(x) = w \cdot \varphi(x) + b \tag{8}$$

dimana simbol $\varphi(x)$ berfungsi untuk memetakan data pelatihan dengan dimensi yang lebih tinggi, agar mudah diproses lebih lanjut. Pada proses ini juga melibatkan metode lagrange multiplier untuk menentukan metode yang terbaik dalam regresi, setelah proses tersebut kemudian diterapkan persamaan (5)

$$\max_{\alpha_i \geq 0} \left[\sum_i \alpha_i - \frac{1}{2} \sum_{i,j} \alpha_i \alpha_j y_i y_j K(x_i, x_j) \right] \tag{9}$$

Dimana $K(x_i, x_j)$, merupakan fungsi kernel.

Ada dua parameter yang terdapat pada algoritma SVM yaitu C dan gamma. Parameter C atau Cost merupakan parameter yang digunakan optimalisasi hasil klasifikasi atau nilai C ini berfungsi untuk menghindari kesalahan klasifikasi di setiap sampel data pelatihan. Nilainya dengan jangkauan 0 sampai 1. Sedangkan parameter gamma digunakan untuk menentukan seberapa jauh pengaruh ketergantungan dari satu sampel dataset pelatihan.

2.1.4 Kesimpulan dan Hasil Penelitian

Pada tahap akhir, kami melakukan perangkuman semua hasil penelitian yang telah dibuat pada makalah ini, serta kemungkinan penelitian selanjutnya yang berkaitan dengan hasil penelitian ini.

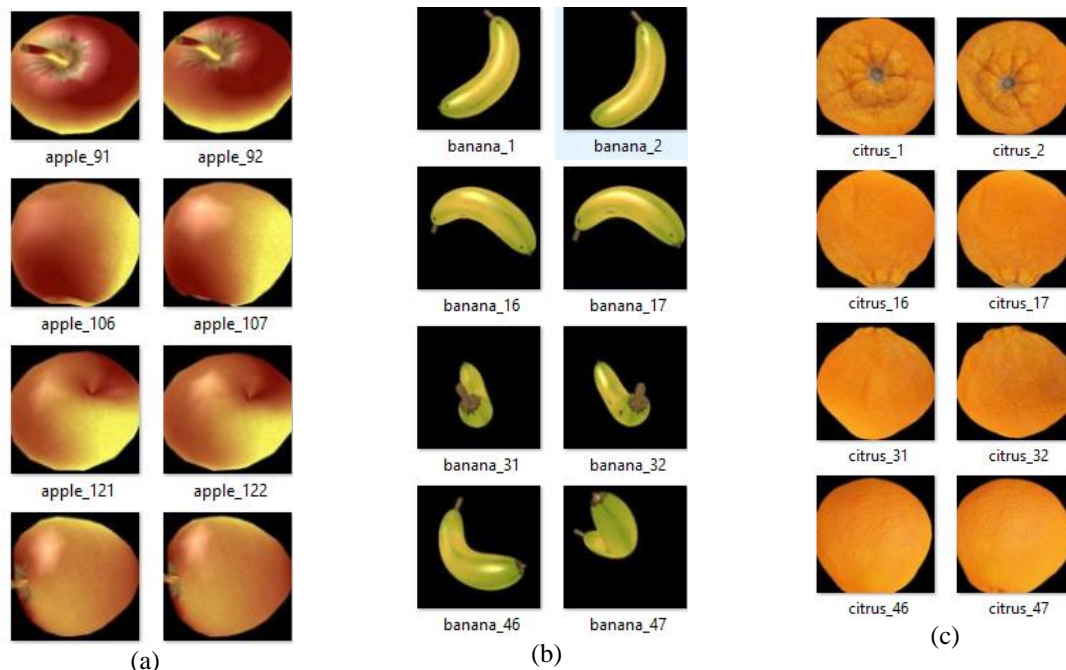
3. HASIL DAN PEMBAHASAN

3.1 Eksperimen

Proses eksperimen penelitian ini menggunakan laptop dengan spesifikasi intel core i7-8750H CPU @2.20GHz, sedangkan memori yang digunakan adalah 16 GB. Sistem operasi menggunakan windows 11 dan pemrograman python versi 3.7.1 yang dilengkapi dengan library untuk k-means dan SVM menggunakan sklearn.

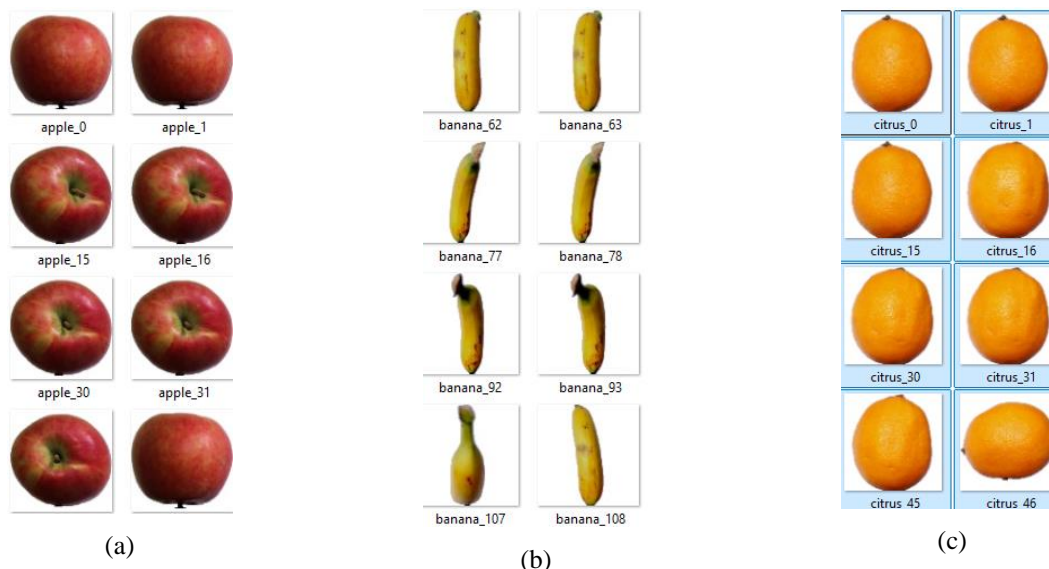
3.1.1 Dataset

Dataset yang digunakan adalah dataset sintesis yang dibangun menggunakan pemrograman java. Dataset ini berbentuk objek tiga dimensi dengan kategori apel, pisang dan jeruk yang visualisasinya seperti Gambar 10 berikut.



Gambar 10. Dataset Apel (a), Pisang(b), dan Jeruk (c)

Pada Gambar 10, dataset yang dikumpulkan tersebut terdiri dari 150 buah apel, 150 buah pisang, dan 150 buah jeruk. Sedangkan dataset uji yang akan digunakan dari penelitian sebelumnya [9] yaitu data nyata hasil capture dengan menggunakan motor DC. Komposisi citra memperlihatkan sisi yang diambil secara acak, untuk apple (a) terlihat bahwa objeknya berupa animasi yang tidak begitu halus teksturnya tetapi masih bisa mewakili sebuah apel, citra buah pisang (b) teksturnya sudah halus dan dapat mewakili objek pisang, sedangkan jeruk (c), teksturnya hamper mirip dengan jeruk sebenarnya karena diambil dari sebuah foto nyata yang kemudian di generate menghasilkan bentuk 3 dimensi.



Gambar 11. Dataset buah nyata, Apel (a), Pisang (b) dan Jeruk (c)

Dataset pada Gambar 11, komposisinya yaitu jumlah 148 buah apel, pisang 152 buah, dan jeruk 166 buah.

3.1.2 Variasi Jumlah Cluster Vocabulary

Pada tahap ini, tujuan utama yang ingin dihasilkan adalah untuk mencari jumlah *cluster vocabulary* yang terbaik dalam meningkatkan akurasi kombinasi algoritma yang ditawarkan, terutama saat ekstraksi fitur menggunakan algoritma SIFT. Karena *dataset* yang digunakan jumlahnya terbatas sebanyak 450 citra untuk proses pelatihan, maka jumlah *cluster* yang akan diobservasi yaitu 5, 6, 7, 8, 9, 10, 11, 12, 13, 14.

3.1.3 Variasi Fungsi Kernel Support Vector Machine (SVM)

Fungsi *kernel* pada algoritma SVM berfungsi untuk mencari antara *hyper-plane* dengan vector. Sedangkan vector yang digunakan adalah *vocabulary* yang dihasilkan dari proses *Bag of feature*. Jenis *kernel* yang akan dilibatkan untuk

mencari efek terbaik terhadap peningkatan akurasi adalah Linier, polinomial, dan Sigmoid. Sedangkan *hyperparameter* yang digunakan untuk semua fungsi *kernel* yaitu $C=1.0$

3.2 Hasil Pengujian

Penelitian ini menggunakan total 450 citra *dataset* sintetik yang digunakan sebagai data pengujian, berisi tiga kelas yaitu apel, pisang, dan jeruk sedangkan data pengujian digunakan dari citra nyata yang diambil dari *dataset* penelitian sebelumnya [9]. Proses pengujian diperlihatkan pada Gambar 12.

```

Input : Training set S (bag of Features) as histogram features; Test instance X. Total number of
models used bag of features = 3 kernels of SVM( linear, polynomial, sigmoid)
Procedure Bag of Features-svm
  Training Process:
    For i ← 1 to 3 do
      Use Histogram features Si number of clusters to
      create the classification models;
    EndFor
  The classification models to testing number of clusters of histogram features X;
  Testing Process:
    For i ← 1 to 3 do
      Classify X;
    EndFor
  Output (majority class of X, accuracy, precision, recall, F-1 score)
EndProcedure
    
```

Gambar 12. Pseudocode untuk proses pengujian *bag of features* menggunakan algoritma SVM

Pada Gambar 12 seluruh hasil klasifikasi menggunakan *library sklearn* yang terintegrasi dengan pemrograman python sehingga dari gambar tersebut dapat dilihat bahwa fungsi *kernel* yang digunakan dalam contoh pengujian pertama ini menggunakan persamaan (9) dan persamaan (5). Sedangkan pengukuran performa klasifikasi tersebut menggunakan persamaan (1) dan (4). Selanjutnya pengujian tersebut dilakukan dengan variasi *hyperparameter* yang terangkum pada Tabel 1, Tabel 2, dan Tabel 3 berikut.

Tabel 1. Variasi jumlah *cluster vocabulary* dengan kernel linier dan $C=1.0$

Jumlah cluster vocabulary	Akurasi (%)	F-1 score (%)
5	51.07	46.46
6	51.28	46.75
7	51.50	44.68
8	54.07	49.58
9	56.86	51.41
10	57.72	52.16
11	60.51	55.29
12	59.65	54.46
13	59.22	53.40
14	58.36	51.42

Berdasarkan hasil pengujian Tabel 1, memperlihatkan bahwa jumlah *cluster vocabulary* paling optimal pada 11 dengan fungsi *kernel* linier, yang mana menghasilkan akurasi paling tinggi 60.51 %, dan pengukuran *F-1 score* 55.29 %. Hal ini menunjukkan bahwa model yang digunakan cukup baik karena secara bersamaan akurasi yang diperoleh cukup bagus dan *F-1 score* juga memperlihatkan keseimbangan dengan akurasi.

Tabel 2. Variasi jumlah *cluster vocabulary* dengan kernel polynomial dan $C=1.0$

Jumlah cluster vocabulary	Akurasi (%)	F-1 score (%)
5	45.92	42.34
6	45.92	41.44
7	36.90	35.24
8	54.07	49.59
9	50.0	48.22
10	57.08	51.73

11	55.15	52.57
12	58.36	56.86
13	49.78	49.21
14	49.35	47.78

Tabel 3. Variasi jumlah *cluster vocabulary* dengan kernel sigmoid dan $C=1.0$

Jumlah cluster vocabulary	Akurasi (%)	F-1 score (%)
5	49.14	44.70
6	51.07	46.45
7	46.56	41.09
8	55.57	49.65
9	65.45	58.45
10	61.80	55.86
11	60.30	54.08
12	58.36	52.57
13	55.36	49.29
14	54.07	50.13

Pada Tabel 2, hasil pengujian memperlihatkan akurasi yang didapatkan paling tinggi adalah 58.36 % dengan F1-score 56.86 % pada jumlah *cluster vocabulary* 12. Terlihat bahwa ketika menggunakan fungsi *kernel* polinomial performa kombinasi algoritma tersebut lebih rendah dibandingkan menggunakan fungsi *kernel* linier. Pada tabel 3 hasil pengujian memperlihatkan kecenderungan lebih bagus dibandingkan dengan kedua fungsi kernel sebelumnya yaitu fungsi *kernel* linier dan fungsi *kernel* polinomial. Pada pengujian menggunakan fungsi *kernel* sigmoid menghasilkan akurasi lumayan tinggi yaitu 65.45 % dengan F1-score mencapai 58.45 % pada jumlah *cluster vocabulary* 9. Sehingga *hyperparameter* yang terbaik untuk digunakan pada algoritma SVM adalah fungsi *kernel* sigmoid.

4. KESIMPULAN

Berdasarkan hasil pengujian dengan observasi jumlah *cluster vocabulary* untuk algoritma SIFT dan BoF dengan variasi fungsi *kernel* pada algoritma SVM untuk melakukan klasifikasi memperlihatkan jumlah *cluster* yang paling optimal untuk kasus klasifikasi buah dengan citra sintetis terletak pada jumlah *cluster* 9 dengan *hyperparameter* $C=1.0$ dan fungsi *kernel* SVM menggunakan fungsi sigmoid. Kombinasi algoritma ini masih mendapatkan akurasi rendah yaitu 65.45% dengan F1-score mencapai 58.45 %, dengan akurasi tersebut memungkinkan bahwa data sintetis buah yang dibuat dapat digunakan untuk melakukan klasifikasi buah yang sebenarnya (*real*). Eksperimen untuk penelitian yang akan datang digunakan *dataset* yang lebih besar lagi dengan menambah varian kelas yang akan dilibatkan serta pengujian variasi parameter akan ditingkatkan lagi sehingga diharapkan dapat meningkatkan kinerja algoritma yang akan diusulkan.

REFERENCES

- [1] A. Bhargava and A. Bansal, "Fruits and vegetables quality evaluation using computer vision: A review," *J. King Saud Univ. - Comput. Inf. Sci.*, vol. 33, no. 3, pp. 243–257, 2021, doi: 10.1016/j.jksuci.2018.06.002.
- [2] N. Ismail and O. A. Malik, "Real-time visual inspection system for grading fruits using computer vision and deep learning techniques," *Inf. Process. Agric.*, vol. 9, no. 1, pp. 24–37, 2022, doi: 10.1016/j.inpa.2021.01.005.
- [3] Y. Gulzar, "Fruit Image Classification Model Based on MobileNetV2 with Deep Transfer Learning Technique," *Sustain.*, vol. 15, no. 3, 2023, doi: 10.3390/su15031906.
- [4] Y. Gurubelli, R. Malmathanraj, and P. Palanisamy, "Texture and Colour Gradient Features for Grade analysis of Pomegranate and Mango Fruits using kernel-SVM Classifiers," *2020 6th Int. Conf. Adv. Comput. Commun. Syst. ICACCS 2020*, no. September 2021, pp. 122–126, 2020, doi: 10.1109/ICACCS48705.2020.9074221.
- [5] A. Hassoun *et al.*, "Implementation of relevant fourth industrial revolution innovations across the supply chain of fruits and vegetables: A short update on Traceability 4.0," *Food Chem.*, vol. 409, no. November 2022, 2023, doi: 10.1016/j.foodchem.2022.135303.
- [6] J. Amin, M. A. Anjum, M. Sharif, S. Kadry, and Y. Nam, "Fruits and vegetable diseases recognition using convolutional neural networks," *Comput. Mater. Contin.*, vol. 70, no. 1, pp. 619–635, 2021, doi: 10.32604/cmc.2022.018562.
- [7] T. T. M. Huynh, T. M. Le, L. T. That, L. Van Tran, and S. V. T. Dao, "A Two-Stage Feature Selection Approach for Fruit Recognition Using Camera Images with Various Machine Learning Classifiers," *IEEE Access*, vol. 10, no. August, pp. 132260–132270, 2022, doi: 10.1109/ACCESS.2022.3227712.
- [8] L. G. Fahad, S. F. Tahir, U. Rasheed, H. Saqib, M. Hassan, and H. Alquhayz, "Fruits and Vegetables Freshness Categorization Using Deep Learning," *Comput. Mater. Contin.*, vol. 71, no. 2, pp. 5083–5098, 2022, doi: 10.32604/cmc.2022.023357.
- [9] H. Mureşan and M. Oltean, "Fruit recognition from images using deep learning," *Acta Univ. Sapientiae, Inform.*, vol. 10, no. 1, pp. 26–42, 2018, doi: 10.2478/ausi-2018-0002.
- [10] H. S. Gill, G. Murugesan, B. S. Khehra, G. S. Sajja, G. Gupta, and A. Bhatt, "Fruit recognition from images using deep



- learning applications,” *Multimed. Tools Appl.*, vol. 81, no. 23, pp. 33269–33290, 2022, doi: 10.1007/s11042-022-12868-2.
- [11] H. H. C. Nguyen, C. Jana, I. M. Hezam, H. P. Hieu, and N. T. Thuy, “Identification of dragon trees and fruits in ham Thuan Bac growing areas, Phan Thiet city, Binh Thuan province, Vietnam,” *Heliyon*, vol. 10, no. 10, p. e31233, 2024, doi: 10.1016/j.heliyon.2024.e31233.
- [12] N. N. A. A. Hamid, R. A. Razali, and Z. Ibrahim, “Comparing bags of features, conventional convolutional neural network and alexnet for fruit recognition,” *Indones. J. Electr. Eng. Comput. Sci.*, vol. 14, no. 1, pp. 333–339, 2019, doi: 10.11591/ijeecs.v14.i1.pp333-339.
- [13] S. K. Behera, A. K. Rath, and P. K. Sethy, “Fruit recognition using support vector machine based on deep features,” *Karbala Int. J. Mod. Sci.*, vol. 6, no. 2, pp. 235–245, 2020, doi: 10.33640/2405-609X.1675.
- [14] Y. Peng *et al.*, “CNN-SVM: A classification method for fruit FL image with the complex background,” *IET Cyber-Physical Syst. Theory Appl.*, vol. 5, no. 2, pp. 181–185, 2020, doi: 10.1049/iet-cps.2019.0069.
- [15] L. Ge *et al.*, “Three dimensional apple tree organs classification and yield estimation algorithm based on multi-features fusion and support vector machine,” *Inf. Process. Agric.*, vol. 9, no. 3, pp. 431–442, 2022, doi: 10.1016/j.inpa.2021.04.011.
- [16] G. Arora, A. K. Dubey, Z. A. Jaffery, and A. Rocha, “Bag of feature and support vector machine based early diagnosis of skin cancer,” *Neural Comput. Appl.*, vol. 34, no. 11, pp. 8385–8392, 2022, doi: 10.1007/s00521-020-05212-y.
- [17] L. Li and M. Iskander, “Comparison of 2D and 3D dynamic image analysis for characterization of natural sands,” *Eng. Geol.*, vol. 290, no. June 2020, p. 106052, 2021, doi: 10.1016/j.enggeo.2021.106052.
- [18] Q. Sun, J. Zheng, M. R. Coop, and F. N. Altuhafi, “Minimum image quality for reliable optical characterizations of soil particle shapes,” *Comput. Geotech.*, vol. 114, no. April, p. 103110, 2019, doi: 10.1016/j.compgeo.2019.103110.
- [19] R. Kurban, “Gaussian of Differences: A Simple and Efficient General Image Fusion Method,” *Entropy*, vol. 25, no. 8, 2023, doi: 10.3390/e25081215.
- [20] R. Azhar, D. Tuwohingide, D. Kamudi, Sarimuddin, and N. Suciati, “Batik Image Classification Using SIFT Feature Extraction, Bag of Features and Support Vector Machine,” *Procedia Comput. Sci.*, vol. 72, pp. 24–30, 2015, doi: 10.1016/j.procs.2015.12.101.