

# Pengenalan Motif Kain Songket Pada Citra Kamera Smartphone Dengan Beragam Sudut Pandang Menggunakan CNN

Muhammad Husein Nashr<sup>1</sup>, Muhammad Fachrurrozi<sup>1\*</sup>, Eni Triningsih<sup>2</sup>, Kanda Januar Miraswan<sup>1</sup>

<sup>1</sup> Teknik Informatika, Universitas Sriwijaya, Palembang, Indonesia

<sup>2</sup> Computer Science Faculty, Universitas Nasional Pasim, Bandung, Indonesia

Email : mfachrz@unsri.ac.id

**Abstrak**— Songket Palembang memiliki motif yang beragam sehingga dibutuhkan mesin pengenal yang dapat membantu orang awam mengenali motif ini. Mesin pengenal harus mampu mengenali motif dengan variasi transformasi *spatial*, *noisedan blur*. Dalam penelitian ini, CNN mampu mengklasifikasi motif songket dengan akurasi 93%. Arsitektur CNN yang digunakan menggunakan 2.22 MB memori GPU saat *inference*. Penggunaan *Dropout* memberikan efek *regularisasi*, yaitu meningkatkan akurasi pada data uji dan penggunaan *momentum* dengan nilai 0.9 mengurangi waktu *training* 2x lebih cepat. *Layer* konvolusi CNN pada penelitian ini tidak dapat mengekstrak fitur penting pembeda antar kelas, tidak seperti *layer* konvolusi CNN *pretrain* yang sudah dilatih dengan dataset yang besar sehingga menghasilkan akurasi 100% untuk klasifikasi songket

**Kata Kunci**— *Songket Pattern Classification, Scale Invariant Feature Transform, Bag of Words, Fully Connected Neural Network, Convolutional Neural Network*

## I. LATAR BELAKANG

Songket Palembang memiliki motif yang beragam, maka dibutuhkan *classifier* atau mesin pengenal yang dapat membantu orang awam mengenali motif-motif ini. Masalah yang muncul adalah citra motif tidak selalu diambil dari sudut pandang yang sama. *Classifier* yang dibuat juga harus bisa menerima input citra kamera *smartphone* agar bisa digunakan dimana saja dan kapan saja. Masalah yang muncul adalah citra juga bisa mengandung *noise* dan *blur* jika diambil didalam ruangan. Hal ini disebabkan oleh sensor kamera *smartphone* yang terlalu kecil. Untuk itu diperlukan penelitian untuk mencari *classifier* yang *robust* terhadap transformasi *spatial*, *noise* dan *blur*.

Mesin pengenal yang dibuat harus mampu mengenali motif walaupun dengan sudut kamera yang berbeda, dengan *noise* ataupun *blur*. Kemampuan ini dibutuhkan karena pengguna tidak selalu menempatkan kamera pada posisi yang sama dan juga tidak selalu mendapatkan cahaya yang cukup untuk kamera *smartphone* yang kecil sehingga menghasilkan *blur*.

Penelitian tentang pengenalan motif kain songket pernah dilakukan oleh Riztyan & Dariska [1] dan Willy dkk [2]. Pada penelitian Riztyan & Dariska, pengujian pengenalan dilakukan dengan variasi rotasi, sedangkan pada Willy dkk, pengujian dilakukan dengan *noise*. Walaupun masing-masing menghasilkan akurasi yang bagus pada pengujian, yakni 90% dan 95%, tetapi *constraint* pengujian masih tidak

cukup. Motif songket tidak selalu diambil dari sudut pandang dan lokasi yang sama, dan secara bersamaan, pencahayaan yang kurang untuk sensor kamera *smartphone* yang kecil membuat citra menjadi *blur & noisy*. Oleh karena itu, penelitian untuk mencari *Classifier* harus *robust* terhadap *constraint* ini masih harus dilakukan.

Penelitian ini menggunakan fitur ekstraktor *Scale Invariant Feature Transform* (SIFT) dalam proses pengenalan motif kain songket. SIFT merupakan fitur ekstraktor yang *invariant* terhadap variasi skala, rotasi, *illuminasi* dan sudut pandang [3]. Fitur ekstraktor ini mampu melakukan klasifikasi citra dengan akurasi 90% pada pengujian terhadap skala, rotasi, *brightness* dan *noise*. [4]. Selain itu, SIFT juga digunakan pada klasifikasi motif kain songket dengan *noise* [2].

Fitur ekstraktor SIFT akan dikombinasikan dengan *feature transformation Bag of Words* (BoW). SIFT dan BoW ini dipakai pada penelitian mengenai pengenalan batik dengan skala dan rotasi [5] dan pengenalan citra dengan *scaling*, rotasi, *brightness changes*, dan *noise* [4], dengan rata-rata akurasi sekitar 90% pada skenario uji. Selain itu algoritma *Feedforward Neural Network* dipakai sebagai *feature classifier* karena dapat mengalahkan SVM dalam mengenali fitur SIFT+BoW [6]. Kombinasi algoritma ini dipilih sebagai *benchmark* pengenalan memiliki performa di beberapa penelitian lain dengan masalah yang sama dengan penelitian ini.

Pada penelitian ini juga dikembangkan arsitektur *Convolutional Neural Network* (CNN) yang dapat mengenali kain songket dengan variasi transformasi *spatial*, *blur* dan *noise*. Keberhasilan metode tradisional dalam menyelesaikan masalah *computer vision* bergantung pada ekstraksi fitur. CNN bisa secara otomatis belajar mengenali fitur suatu domain permasalahan spesifik, tanpa perlu membuat fitur ekstraktor *handcrafted* untuk masalah tersebut [7]. CNN menunjukkan hasil *state of the art* pada banyak *benchmark* klasifikasi yang menantang, mendominasi metode lain dalam kompetisi klasifikasi, bahkan mengungguli performa manusia dalam melakukan klasifikasi satu label [8].

Pengenalan motif tradisional dengan CNN dilakukan oleh Yohanes dkk pada motif batik [6]. Penelitian tersebut menggunakan model VGG16 yang sudah di-*pretrained* dengan dataset ImageNet, teknik ini disebut *network based transfer learning*. CNN yang digunakan mampu mengenali motif batik dengan akurasi sebesar 89±7%. Pada pengujian rotasi dan skala CNN mampu dikalahkan metode lain, tetapi

perlu dicatat bahwa Layer *convolution* CNN tersebut tidak dilatih dengan dataset motif batik dengan variasi skala dan rotasi. Di sisi lain pelatihan ulang VGG16 juga tidak menjamin performa yang bagus. Pada penelitian Teny Handhayani dkk, arsitektur CNN dengan 3-layer Convolution & Max Pooling mampu mengalahkan VGG16 [9]. Untuk itu pada penelitian ini, dikembangkan arsitektur dimulai dari CNN 3-layer.

Selain itu, CNN yang dikembangkan dilengkapi dengan modul *Spatial Transformer Network* (STN). Modul yang dikembangkan oleh DeepMind pada tahun 2015 ini dipilih karena berhasil mencapai performa *state of the art* dalam masalah variasi transformasi spasial [10]. Selanjutnya digunakan juga *data augmentation* untuk melengkapi variasi transformasi spasial, blur dan noise pada data training. Data training yang sudah di augmentasi ini membuat CNN robust terhadap variasi translasi [11] dan noise [12].

Berdasarkan uraian diatas, performa SIFT+BoW+FNN dalam mengenali motif kain songket akan dijadikan acuan (*benchmark*). *Convolutional Neural Network* dikembangkan untuk mengenali motif kain songket dengan variasi transformasi *spatial*, *blur* dan *noise*. Arsitektur CNN dikembangkan memiliki 3-layer *convolutional* dan menggunakan modul *Spatial Transformer Network* (STN) untuk mengatasi variasi *spatial*. *Data Augmentation* juga digunakan untuk meningkatkan variance CNN terhadap translasi & noise.

## II. STUDI PUSTAKA

### A. Motif Kain Songket

Asal usul songket tidak begitu jelas, tetapi teori populer menyatakkan bahwa songket disebarakan sebagai pakaian keluarga kerajaan. Banyaknya pernikahan campuran dan perdagangan meningkatkan jenis kain songket. Ada motif kain yang dibuat dengan tema cina dsb [13]. Songket pada saat itu lebih populer karena bahan baku pembuatan lebih mudah didapatkan daripada proses pembuatan batik. Kain ini ditunen menggunakan tangan dengan benang emas, perak dan kapas putih serta pada umumnya dikenakan pada acara resmi.

Kain songket memiliki bentuk geometris berulang. Dalam satu jenis kain songket terdapat beberapa bagian motif pembentuk. Motif pembentuk ini antara lain motif songket (bunga), tumpal, kandang, dan bordir [13]. Motif-motif bisa dibedakan berdasarkan bentuk motif, kepadatan motif, jumlah warna dan posisi motif pada songket.

### B. Citra

Citra merupakan gambar pada bidang dua dimensi. Secara matematis citra merupakan fungsi *continuous* dari intensitas cahaya pada bidang dua dimensi. Sumber cahaya yang memantul dari objek ditangkap oleh *alat optik* seperti mata, kamera atau *scanner*, sehingga objek yang disebut citra tersebut terekam [14].

Citra digital merupakan citra dengan resolusi spasial dan nilai intensitas diskrit. Bagian terkecil dari citra digital

dinamakan piksel. Piksel memiliki koordinat dan nilai intensitas. Citra digital direpresentasikan sebagai matriks 3 dimensi. Dimensi 1 dan 2 mewakili koordinat *spatial* suatu piksel. Sedangkan dimensi 3 merupakan dimensi warna dengan panjang 3 unit, mewakili warna merah (R), hijau (G), dan biru (B).

$$R = \begin{bmatrix} f(0,0,0) & \dots & f(0,N-1,0) \\ \vdots & \ddots & \vdots \\ f(M-1,0,0) & \dots & f(M-1,N-1,0) \end{bmatrix} \quad (1)$$

$$G = \begin{bmatrix} f(0,0,1) & \dots & f(0,N-1,1) \\ \vdots & \ddots & \vdots \\ f(M-1,0,1) & \dots & f(M-1,N-1,1) \end{bmatrix} \quad (2)$$

$$B = \begin{bmatrix} f(0,0,2) & \dots & f(0,N-1,2) \\ \vdots & \ddots & \vdots \\ f(M-1,0,2) & \dots & f(M-1,N-1,2) \end{bmatrix} \quad (3)$$

Dimana  $f(x, y, c)$  adalah intensitas pada piksel dengan koordinat spasial  $x, y$  dan warna  $c$ .  $M$  dan  $N$  mewakili resolusi citra digital vertikal dan horizontal (atau sebaliknya, sesuai konvensi).

Citra *grayscale* merupakan jenis citra dengan 1 warna. Citra RGB dapat diproses menjadi citra grayscale dengan menjumlahkan setiap warna menggunakan bobot tertentu. Pada penelitian ini digunakan formula CCIR yang populer digunakan dalam mengubah citra RGB ke *grayscale*. Dibawah ini merupakan formula tersebut.

$$Y(x, y) = 0.299 * R(x, y) + 0.587 * G(x, y) + 0.114 * B(x, y) \quad (4)$$

### C. Klasifikasi Citra dan Ekstraksi Fitur

Klasifikasi citra merupakan masalah memberikan label dari sebuah himpunan kategori tertentu kepada sebuah citra masukan. Mula-mula citra di praproses sesuai kebutuhan fitur ekstraktor atau masalah. Selanjutnya dilakukan ekstraksi fitur. Kemudian, fitur-fitur digunakan untuk melatih *classifier* dan didapatkan sebuah model. Model ini digunakan untuk memberikan label terhadap suatu fitur baru.

Ekstraksi fitur dilakukan untuk mendapatkan informasi paling relevan dari citra dalam dimensi yang lebih kecil [15], sehingga klasifikasi dapat dilakukan dengan baik. Sebuah fitur ekstraktor dibuat secara manual (*handcrafted*) untuk dipakai pada masalah tertentu. Salah satu pendekatan fitur ekstraktor adalah dengan mentransformasi citra menjadi vektor fitur yang merupakan koefisien-koefisien deret ekspansi. Fitur ekstraktor ini antara lain *Wavelet Transform* dan *Moment*. Pendekatan lain adalah lokal fitur dengan *keypoint* seperti SIFT. Ada juga fitur ekstraktor yang secara otomatis belajar mengekstraksi fitur-fitur dengan baik pada sebuah domain, seperti layer *convolutional* pada CNN.

Pada SIFT, sekumpulan tepi dicari dengan metode *Difference of Gaussian* (DOG). Tepi-tepi yang curam akan diambil dan dijadikan *keypoints*. Untuk setiap *keypoints* akan ditentukan 128 nilai descriptor. Pada akhirnya SIFT akan

menghasilkan fitur sepanjang banyaknya  $keypoints * 128$  deskriptor (Histogram N ukuran  $4 \times 4$  dengan  $bins$  rotasi R sebanyak 8) untuk setiap gambar [3]. Penjelasan lebih lanjut mengenai SIFT dijelaskan pada subbab 2.6.

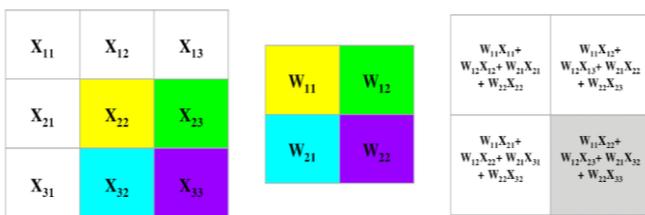
Fitur-fitur yang didapatkan dari fitur ekstraktor SIFT tidak dapat langsung digunakan untuk klasifikasi. Hal ini dikarenakan jumlah fitur pada setiap citra tidak sama, dikarenakan  $keypoints$  yang ditemukan bisa saja berbeda pada tiap citra [6]. BoW mentransformasi fitur agar memiliki panjang yang sama.

BoW bekerja dengan mengelompokkan (*clustering*)  $keypoints$  yang ditemukan pada seluruh citra berdasarkan deskriptornya menggunakan algoritma K-Means. Setelah kelompok-kelompok dengan panjang K didapatkan, dihitung jumlah deskriptor yang masuk pada setiap kelompok pada sebuah citra. Pada akhirnya didapatkan *Bag of Words* [16] yang merupakan fitur baru dengan panjang K untuk setiap citra. Jumlah fitur K yang digunakan pada penelitian ini yaitu 2800 sesuai pada penelitian Gultom dkk [6].

Setelah mendapatkan fitur dengan panjang K pada setiap citra, fitur-fitur tersebut kemudian diklasifikasi dengan FNN [6]. FNN yang digunakan memiliki *input* dengan K unit neuron, 1 *hidden layer* dengan 1000 unit *neuron* dengan aktivasi ReLu [6]. *Output* FNN merupakan vektor dengan panjang C, dimana C adalah jumlah kelas. Elemen-elemen vector ini merupakan nilai probabilitas prediksi setiap kelas.

#### D. Convolutional Neural Network

*Convolutional Neural Network* muncul karena *Neural network* biasa tidak bisa menerima input berupa gambar dengan ukuran besar. Ukuran  $200 \times 200 \times 3$  pixel saja akan membutuhkan 120.000 bobot. Bobot yang banyak bukan hanya memakan banyak komputasi, tetapi juga menimbulkan overfitting [17]. Untuk mengatasi masalah tersebut CNN memakai operasi konvolusi. Misal seperti pada gambar 1, sebuah neuron 2D (atau 3D) dibuat dengan ukuran  $2 \times 2$ . Selanjutnya setiap bobot neuron dikalikan dengan setiap nilai di sub *input* ukuran  $2 \times 2$  dan semuanya dijumlahkan. Selanjutnya neuron digeser ke  $2 \times 2$  sub *input* berikutnya untuk dilakukan operasi yang sama. Dengan cara ini jumlah bobot tidak harus mengikuti ukuran gambar dan tentunya akan jauh lebih sedikit. Kemudian *classifier* ini juga menjadi *translation invariant*, artinya tetap bisa mengenali pola walaupun muncul di bagian *input* yang tidak pernah dijumpai di data training. Pada akhirnya mengurangi overfitting dan membuat network lebih *robust* dan *general*.



Gambar 1. Operasi Konvolusi pada Fitur Input

Klasifikasi motif kain tradisional dengan CNN dilakukan dengan 2 tahap. Tahapan tersebut secara terurut adalah

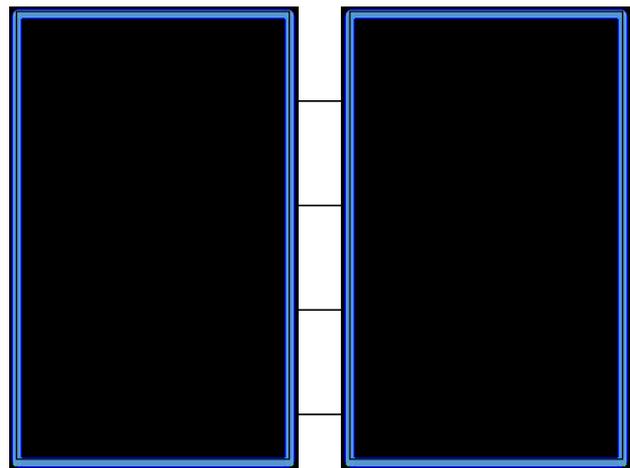
praproses citra dan klasifikasi dengan CNN. Metode CNN tidak membutuhkan fitur ekstraksi, karena sudah digantikan dengan layer *convolution* pada CNN [7].

Tahapan praproses citra pada metode ini melibatkan *resize* atau *down/upsampling* citra dan standarisasi nilai intensitas citra. Semua citra input perlu di *resize* ke dalam ukuran yang sama agar sesuai dengan spesifikasi arsitektur CNN. Ukuran citra yang digunakan pada penelitian ini yaitu  $200 \times 200$ px. Sedangkan standarisasi dilakukan agar CNN dapat bekerja dengan baik (menghindari masalah *dead neuron*) dan mempercepat *training*. Pada penelitian ini juga dilakukan *grayscale*.

Citra yang sudah dipraproses kemudian dikelompokkan dalam suatu *batch* berisi n citra. Pada penelitian ini digunakan 64 buah citra dalam 1 batch. Kemudian *batch-batch* ini dimasukkan dalam CNN sebagai fitur *input* berukuran  $64 \times 200 \times 200 \times 3$ . Setelah *input* diproses, CNN akan mengeluarkan input dengan besar  $n \times \text{banyak\_kelas}$ , dimana setiap elemen merupakan nilai probabilitas suatu kelas.

### III. METODOLOGI PENELITIAN

Gambar 2 merupakan tahapan yang akan dilakukan pada penelitian ini.



Gambar 2. Kerangka Kerja Penelitian

#### 1. Literature Review

Tahap ini dilakukan untuk mempelajari tentang metode *convolutional neural network*. Hasil yang didapat dalam fase ini adalah pemahaman tentang CNN, *best-practices* dalam membangun arsitekturnya dan pemahaman tentang proses training dan *inference* CNN.

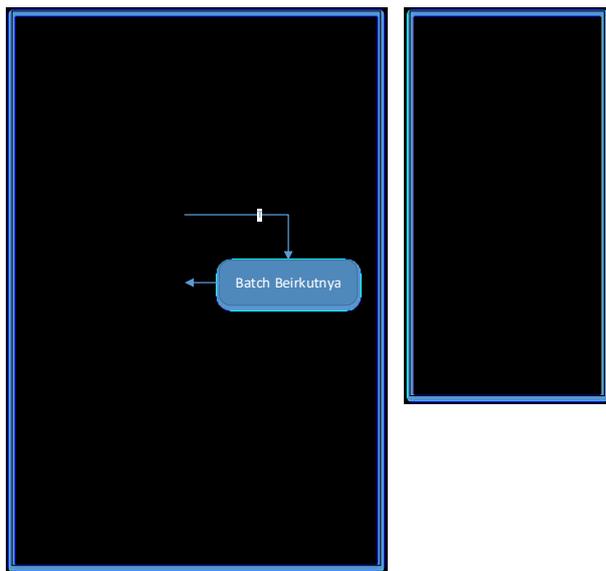
#### 2. Pengumpulan Data

Objek di dalam penelitian ini adalah data citra motif kain songket. Data yang dikumpulkan merupakan data primer dengan metode pengumpulan dokumentasi. Dokumentasi ini berupa pengumpulan data citra kain songket dari suplier kain songket yang ada di Palembang. Terdapat 12 jenis motif kain songket yang akan didokumentasi. Motif kain ini yaitu cantik manis: bunga mawar, bintang, tampuk manggis; lepus: makan ulat, naga, bintang naman, bintang berkait, bintang mawar, pulir, bintang bunga jatuh; cino: bunga mawar dan bunga naman. Untuk dataset training dan validasi masing memiliki 5 dan 1 contoh unik

motif kain songket. Motif kain songket ini dipotret dengan spesifikasi menggunakan kamera DSLR dengan sensor APS-C 12MP, lensa dengan focal length 35mm, pencahayaan yang cukup dan posisi kamera tegak lurus terhadap kain dengan jarak ke kain songket menyesuaikan, sampai semua bagian motif songket masuk dalam *frame*.

### 3. Pembuatan Perangkat Lunak

Pada tahap ini perangkat lunak dikembangkan untuk mendukung pengujian. Pengembangan dilakukan dengan metode RUP. Perangkat lunak yang akan dikembangkan yaitu sistem pengenalan dengan CNN menggunakan python sebagai *backend*; aplikasi GUI *desktop* untuk training dan pengujian; dan aplikasi android untuk demonstrasi kegunaan pengenal motif kain songket. Gambar 3 merupakan alur proses dalam proses *training* dan *inference* dengan menggunakan CNN.



Gambar 3. Activity Diagram untuk Proses *Training* dan *Inference*

### 4. Pengujian dan Analisis Hasil

Pada penelitian ini metode pengenalan diuji dengan mengukur akurasi dalam mengenali kain songket. Skenario prngujian dilakukan untuk menguji metode dalam mengenali pola songket dengan translasi. Data *training* dan data *testing* didapatkan dari memecahkan citra *raw* kedalam beberapa bagian. Satu bagian sebagai data uji, bagian lainnya sebagai data training. Hasil pengujian yang didapatkan selanjutnya dianalisis. Analisis dilakukan pada keberhasilan maupun kegagalan algoritma dalam mengklasifikasi motif songket. Sebab-sebab keberhasilan metode diharapkan bisa ditemukan dan bisa di generalisasi. Disisi lain, sebab-sebab kegagalan metode bisa dihindari dipenelitian selanjutnya.

## IV. HASIL DAN PEMBAHASAN

Pada bagian ini akan dilakukan pengujian terhadap perangkat lunak klasifikasi yang dibuat. Pengujian ini berupa akurasi CNN dalam mengenali motif kain songket dengan transformasi translasi. Pada skenario pertama, metode CNN diuji dalam mengenali motif kain songket dengan variasi translasi. Metode CNN diujikan dengan citra yang memiliki motif ditempat yang berbeda dengan lokasi motif yang ada di

*dataset training*. Dilakukan beberapa kali percobaan, hasil terbaiknya metode CNN mampu mengenali 14 dari 15 (93%) data uji setelah. Gambar 4 merupakan citra dengan prediksi yang benar.



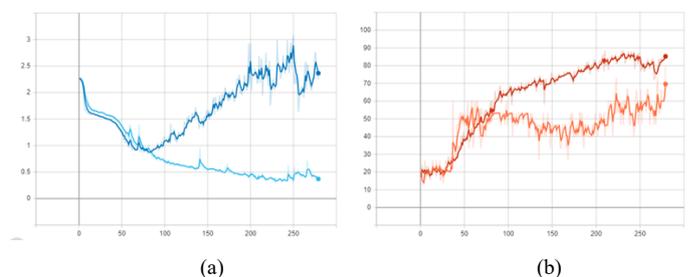
Gambar 4. Citra dengan Prediksi yang Benar

Model CNN yang dilatih pada percobaan-percobaan tersebut dilatih antara 200 – 400 *epoch* dari 250 data training dengan mini-batch ukuran 64. Dari banyak percobaan, rata-rata *epoch* saat model *overfit* sekitar 80 *epoch*. Namun percobaan tidak dihentikan karena bias pada data *train* dan *variance* pada data uji masih besar, dengan akurasi sekitar 50% untuk data train dan 50% untuk data uji.

Berdasarkan hasil percobaan yang dilakukan dapat disimpulkan bahwa motif lepus lebih sulit dikenali dibandingkan motif cantik manis. Terlalu rapatnya bunga bisa membuat filter konvolusi sulit membedakan varian antar kelas. Hal ini bisa diperburuk oleh variasi posisi pada skenario ini. CNN juga dapat membedakan motif rapat dan motif jarang, dilihat dari tidak adanya motif lepus yang diprediksi menjadi cantik manis atau sebaliknya.

Pengenalan motif kain songket tidak perlu arsitektur yang terlalu besar (hanya 120.000 parameter, atau 2.22 MB untuk setiap input). Pada pengujian juga ditemukan bahwa jumlah parameter yang terlalu besar akan membuat model *overfitting* pada training (*variance* terlalu tinggi). Hal ini bisa dikurangi dengan *dropout*. Nilai probabilitas *dropout* yang tinggi juga menyebabkan *loss* menjadi tidak stabil saat proses training. Hal ini bisa disebabkan karena kurangnya jumlah parameter yang aktif pada setiap iterasi yang bisa memodelkan data latih dengan baik

Pada saat pengujian, model mengalami *overfitting* terlalu cepat. Pada gambar 5 dapat dilihat bahwa nilai *loss* uji meningkat dan menjauh dari *loss* latih secara signifikan mulai dari *epoch* 80 saat akurasi data latih masih rendah. Hal ini menunjukkan besarnya varian antara data latih dan data uji. Solusi dari masalah ini adalah menggunakan lebih banyak augmentasi atau mengumpulkan lebih banyak data.



Gambar 5. (a) *Loss* dan (b) Akurasi Hasil Klasifikasi

## V. KESIMPULAN

Pada penelitian ini telah dibangun sebuah perangkat lunak untuk klasifikasi motif songket. Berdasarkan hasil percobaan

yang dilakukan dapat disimpulkan bahwa arsitektur CNN dapat mengenali dengan baik data uji dengan variasi translasi. Motif songket yang rapat (lepus) lebih sulit dikenali karena filter konvolusi sulit mengenali varian antar kelas, yang diperparah dengan variasi translasi. Selain itu ditemukan bahwa pengenalan tidak memerlukan arsitektur CNN yang besar. Arsitektur yang digunakan memiliki 100.000 parameter dan memakan ruang 2.22 MB pada memori. Penggunaan *dropout* dapat meningkatkan akurasi pada data uji. Tetapi *dropout* yang terlalu tinggi dapat mengurangi parameter secara signifikan dan mengurangi kemampuan model untuk fitting data training. *Loss* uji menurun pada saat akurasi model CNN pada data latih masih terlalu tinggi. Hal ini berarti data training dan uji masih memiliki *variance* yang tinggi sehingga dibutuhkan data tambahan berupa *data syntetic* dengan augmentation atau data real pada skenario pengujian.

#### DAFTAR PUSTAKA

- [1] A. Riztyan and R. B. Dariska, "Analisis Pengenalan Motif Songket Palembang Menggunakan Algoritma Propagasi Balik," pp. 1–8, 2013.
- [2] D. Willy, A. Noviyanto, and A. M. Arymurthy, "Evaluation of SIFT and SURF features in the songket recognition," *2013 Int. Conf. Adv. Comput. Sci. Inf. Syst. ICACSIS 2013*, pp. 393–396, 2013.
- [3] D. G. Lowe, "Distinctive Image Features from Scale-Invariant Keypoints," *Int. J. Comput. Vis.*, pp. 1–28, 2004.
- [4] Q. Li and X. Wang, "Image Classification Based on SIFT and SVM," *Proc. - 17th IEEE/ACIS Int. Conf. Comput. Inf. Sci. ICIS 2018*, no. 1, pp. 762–765, 2018.
- [5] 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.
- [6] Y. Gultom, A. M. Arymurthy, and R. J. Masikome, "Batik Classification using Deep Convolutional Network Transfer Learning," *J. Ilmu Komput. dan Inf.*, vol. 11, no. 2, p. 59, 2018.
- [7] N. Aloysius and M. Geetha, "A review on deep convolutional neural networks," *Proc. 2017 IEEE Int. Conf. Commun. Signal Process. ICCSP 2017*, vol. 2018-Janua, pp. 588–592, 2018.
- [8] W. Rawat, "Deep Convolutional Neural Networks for Image Classification: A Comprehensive Review," vol. 2449, pp. 2352–2449, 2017.
- [9] T. Handhayani, J. Hendryli, and L. Hiryanto, "Comparison of Shallow and Deep Learning Models for Classification of Lasem Batik Patterns," pp. 11–16, 2017.
- [10] M. Jaderberg, K. Simonyan, A. Zisserman, and K. Kavukcuoglu, "Spatial Transformer Networks," p. 9, 2015.
- [11] E. Kauderer-Abrams, "Quantifying Translation-Invariance in Convolutional Neural Networks," 2017.
- [12] N. Tiago S., P. da C. Gabriel B., C. Welinton A., and P. Moacir, "Deep Convolutional Neural Networks and Noisy Images," *Lect. Notes Comput. Sci.*, vol. 1, pp. 416–424, 2018.
- [13] A. Hamzah, "Wawancara Megenai Songket Palembang." Palembang, 2019.
- [14] R. Munir, "Pengolahan Citra Digital," *Inform. Bandung*, pp. 229–236, 2002.
- [15] G. Kumar and P. K. Bhatia, "A detailed review of feature extraction in image processing systems," *Int. Conf. Adv. Comput. Commun. Technol. ACCT*, pp. 5–12, 2014.
- [16] Y. Zhang, R. Jin, and Z. H. Zhou, "Understanding bag-of-words model: A statistical framework," *Int. J. Mach. Learn. Cybern.*, vol. 1, no. 1–4, pp. 43–52, 2010.
- [17] C. Zhang, S. Bengio, M. Hardt, B. Recht, and O. Vinyals, "Understanding deep learning requires rethinking generalization," 2016.