

Tomo-e Gozen サーベイにおける画像認識を用いた超新星
の検出

浜崎 凌

March 2020

概要

時間領域天文学は、天体の時間変動に関する天文学である。この分野が取り扱う天体は、超新星や激変星、星のフレア、活動銀河中心核などの突発天体が含まれる。これら突発天体のサーベイでは、一度で広範囲の空の領域をカバーし、かつ素早く観測を行う必要がある。

東京大学木曾観測所では、直径9度の視野を持つ105cm シュミット望遠鏡に84枚のCMOS センサ(合計視野20平方度)を搭載した可視光の広視野サーベイのプロジェクト「Tomo-e Gozen」が進行している。そのプロジェクトの一環として、毎晩数時間おきに夜空を巡回し観測することで超新星爆発の瞬間を狙っている。新しく撮影された画像に対して以前撮影された同じ領域の画像との差分をとることで、その時刻間で発生した突発天体を素早く見つけ出す。しかし実際には、差分処理に失敗した天体などのArtifact 天体も写ってしまい、超新星を含めたこれらの天体の数は一夜あたり~100,000 天体も検出される。

検出された天体から人の目による超新星の迅速な発見をサポートするために、画像認識の一手法である畳み込みニューラルネットワークによって超新星の候補を選出するシステムを開発した。分類性能の高い画像認識を行うには、Model に学習させる超新星の画像データを数万天体用意する必要がある。そこで、観測された恒星の画像から得た統計的な光の拡がりを用い、超新星の画像データを人工的に作成し学習データとした。また、Model の評価には、既に報告されている突発天体のうち Tomo-e の視野内に入っていた天体と Artifact 天体を用いた。

Model の評価の結果、分類性能は天体の信号対雑音比 (SNR) に依存しており、 $SNR > 20$ の天体については分類性能が高いことが分かった。また、この Model によって見つかる新しい本物の突発天体について、 $SNR > 80$ では1晩に約300天体に1つの本物の突発天体が見つかることが分かった。

目次

1	超新星	4
1.1	超新星とは	4
1.2	重力崩壊型超新星のショックブレイクアウト	5
1.3	本稿の構成	6
2	Transient Survey における機械学習の活用	7
2.1	REAL-BOGUS Classification	7
2.2	Random Forest による分類	8
2.3	特徴量が Model の性能に与える影響	9
2.4	特徴量として画像を用いる	10
2.5	Random Forest と CNN の分類性能の比較	11
3	Tomo-e Supernova Survey	13
3.1	天体の検出	14
3.2	天体の検出における初期処理について	16
3.3	画像認識による天体の検出	17
4	学習データ	23
4.1	REAL: Artificial Real	23
4.2	BOGUS: Artifact	26
4.3	評価に用いる指標	26
5	Artificial Real を用いた評価	28
5.1	テストデータについて	28
5.2	Model の ROC curve	28
5.3	SNR について	29
5.4	Model の SNR 依存性	29
6	Real Transient を用いた評価	31
6.1	Real Transient のテストデータ	31
6.2	CNN Model の有無による TPR と FPR の変化	32
6.3	学習データとテストデータに紛れた天体について	34
6.4	$CNN Probability$ の $Threshold$ の変化に対する性能の依存性	36
6.5	Model の SNR 依存性	37
7	まとめ	41
付録 A	Machine Learning のスクリプト	45

1 超新星

1.1 超新星とは

超新星とは大質量星がその進化の最後に起こす爆発現象のことである。その爆発の運動エネルギーは 10^{51} erg、明るさは光度 $\sim 10^{10} L_{\odot}$ にも達し、銀河全体の光度に匹敵する。超新星は異なる仕組みで引き起こされることが知られており、そのスペクトルと光度曲線の特徴を組み合わせで分類される。

超新星爆発の種類	スペクトルの特徴			光度曲線の特徴
	水素	ケイ素	ヘリウム	
Ia	×	○		プラトー有 単調減光
Ib	×	×	○	
Ic	×	×	×	
IIp	○			
IIl	○			
IIin	○ (細い輝線)			

表1 超新星の分類 (山田章一 (著), 2016, ”新天文ライブラリー第4巻超新星”, 日本評論社 (21))

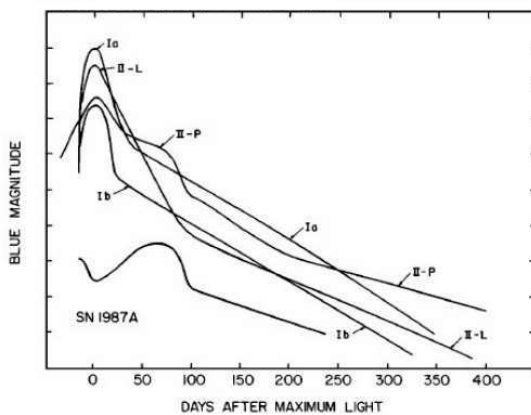


図1 超新星のタイプごとの光度曲線。IIp 型には光度曲線に時間的に光度が変化しない時期 (プラトー) がみられる。(Filippenko, A. V. 1997, ” Optical Spectra of Supernovae” (9))

超新星をスペクトルと光度曲線の特徴から分類したものを表 1 に示す。ただし、中間的な特徴を持つ超新星も存在する。表中のスペクトルの特徴は、最大光度時のスペクトルに水素、ケイ素、ヘリウムの吸収線がみられるかを示している。水素の吸収線がみられるものを I 型、みられないものを II 型と呼ぶ。I 型のうち、ケイ素の吸収線がみられるものは Ia 型、ヘリウムの吸収線がみられるものは Ib 型、いずれの吸収線もみられないものを Ic 型と呼ぶ。Ia 型超新星は白色矮星の核融合反応の暴走に起因する爆発であると考えられている。Ib 型、Ic 型の超新星爆発を起こす元の星はウォルフ・レイエ星であると考えられている。ウォルフ・レイエ星とは、赤色超巨星の段階で自身を構成している表層の物質が外に流れ出た星である。Ib 型は、ヘリウムの外層が

ないウォルフ・レイエ星の重力崩壊による爆発と考えられており、Ic型は水素、ヘリウムの外層がないウォルフ・レイエ星の重力崩壊による爆発と考えられている。II型超新星は水素の吸収線がみられることから水素の外層を残した状態で重力崩壊を起こしたものと考えられている。II_n型にはスペクトルの特徴として他の種類よりも非常に狭い帯域の水素の輝線がみられる。光度曲線は、図1のように星の明るさの時間変化を示した曲線である。図中IIP型には、光度が時間的に変化しない時期であるプラトーがみられる。IIL型はIIP型のよようにプラトーがみられず単調減少していく。

1.2 重力崩壊型超新星のショックブレイクアウト

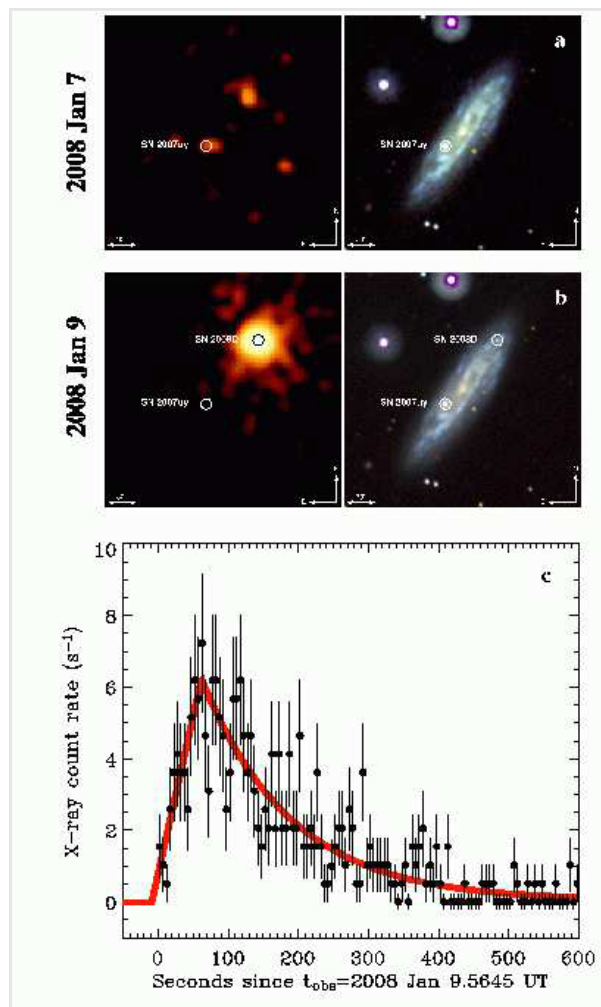


図2 SN 2008DのショックブレイクアウトのX線観測。上2つの画像は超新星爆発前後の図。一番下の図はLight curveを表す。ショックブレイクアウトの継続時間が数百秒であったことが分かる。(Soderberg, A. M., et al. 2008, "An extremely luminous X-ray outburst at the birth of a supernova" (17))

前節で述べたように、Ia型を除く超新星は、星の重力崩壊に起因する爆発と考えられている。重力崩壊型超新星は、コアが重力崩壊を起こしその周りのガスの落下を跳ね返すことで衝撃波が発生する。しかし、星のガ

スが不透明なため、その衝撃波は星の表面に到達するまで外側からは見えない。そして、その星の表面を突き抜けた瞬間に明るく光り始める。この瞬間を「ショックブレイクアウト」と呼ぶ。図 2 は、2008 年に Swift 衛星により発見された Ib 型超新星 SN 2008D の画像と Light curve である。この観測は別のターゲットの観測中に偶然発見された。Light curve は、ショックブレイクアウトを観測したもので、その継続時間は数百秒ほどであった。このようにショックブレイクアウトは明るさがピークに達してから数分～数時間でピーク時の 1/10 の光度まで暗くなり、継続時間が非常に短い。

超新星爆発が起こる直前の星周辺の状態は理論的にも観測的にもほとんど分かっておらず、ショックブレイクアウトの観測は、そういった超新星爆発のメカニズムの解明につながる。

1.3 本稿の構成

本研究では、Tomo-e Supernova Survey における超新星検出のための手法の開発とその結果について述べる。2 章では、過去に行われてきた超新星のサーベイについて、そして 3 章で Tomo-e Gozen サーベイについて述べる。また、天体の検出における問題点とその解決策である画像認識など、天体の検出の方法についても述べる。4 章では、画像認識を行う Model の学習データについて述べる。5 章と 6 章では、4 章で学習させた Model の性能評価を行う。5 章では Artificial Real を用いた評価を行い、6 章では Transient を用いた評価を行う。そして最後に本稿のまとめを 7 章で述べる。

2 Transient Survey における機械学習の活用

時間領域天文学は、天体の時間変動に関する天文学である。この分野が取り扱う天体は、超新星や激変星、星のフレア、活動銀河中心核などの Transient 天体が含まれる。これらの天体はいつどこで現れるか予測することが困難である。また、天体の光度変化を検出するためには、その変化を捉えられる適切な時間間隔での観測が重要である。これらの要件を満たすためには、一度で広範囲の空の領域をカバーし、かつ素早く観測を行う必要がある。Transient 天体のサーベイでは、効率よく天体を探すために、同じ領域の以前の観測画像との差分をとり、その差分画像から天体の検出が行われる。しかし、広範囲の空の領域の観測から検出されるため、1 晩で見つかるこれらの天体の数は非常に多い。差分画像から検出された天体は、不完全な減算処理によるもの、大気の影響によるものなどで生まれる天体 (Artifact 天体) が大半を占める。

2.1 REAL-BOGUS Classification

Transient 天体のサーベイでは、大多数の Artifact 天体から少数の Transient 天体を素早く発見するために機械学習が用いられてきた。機械学習とは、機械的に分類や回帰を行う学習である。これらのサーベイでは、教師データありの学習手法が採用されてきた。教師データありの学習とは、事前に Transient 天体 (REAL) と Artificial 天体 (BOGUS) のデータから得られた特徴量によって Model を学習する。そして学習させた Model を用いて、REAL か BOGUS か不明な天体に対して分類を行う。

これまで Transient 天体のサーベイにおける機械学習の活用は、主な流れとして Random Forest から畳み込みニューラルネットワーク (CNN) へと遷移してきた。Random Forest は多数の決定木 (ツリー構造の分類モデル) から結果を統合する手法であり、天体から得られる情報を特徴量として用いられてきた。CNN は、画像に潜む空間的情報から回帰や分類を行う手法であり、Model が画像から特徴量を抽出する。

Model の評価には True Positive Rate (TPR) と False Positive Rate (FPR) が頻出する。それぞれ、全 REAL のうちどれだけ拾うことができるかの割合、全 BOGUS のうちどれだけ間違っ拾ってしまうかの割合を示す。すなわち、TPR が高く FPR が低いほど分類性能の良い Model である。詳しくは後述する。

2.2 Random Forest による分類

CNN が用いられる以前は、機械学習の Model に与える特徴量を決めて学習させてきた。Palomar Transient Factory(PTF) は、CCD チップを搭載した Palomar 48 インチ望遠鏡を用いて 2009 年から 2012 年まで行われた Transient サーベイの一つである。一晩で検出される天体は 10^6 にも達する可能性があり、そのほとんどは Artifact 天体である。PTF では、これらの Artifact 天体を効率よく除くために Random Forest を用いた。Model の学習には REAL として Transient 天体を 14,781、BOGUS として Artifact 天体を 63,667 の数を用いた。この Random Forest による最終的な Model の性能は (TPR, FPR)=(91.4%, 1%) であった。

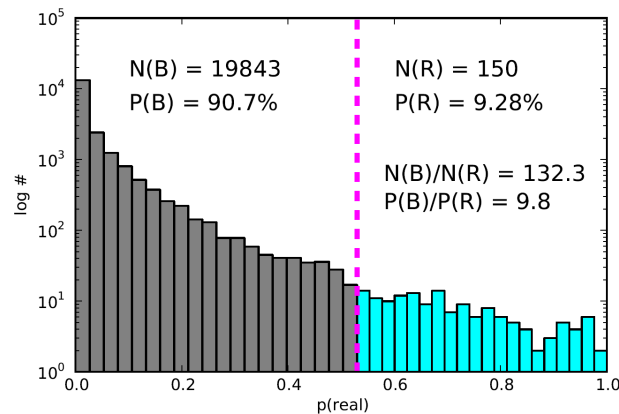


図3 新しく観測されたデータ 20,000 天体に対する Model の予測結果の度数分布。図中赤色の破線は Transient 天体の判定の基準であり、この線よりも左側であれば Artifact、右側であれば Transient として分類される。Artifact と分類された数は $N(B)$ 、Transient と分類された数は $N(R)$ である。また Transient と分類されたデータのうち Artifact であると予想される割合を $P(B)$ 、本物の Transient であると予想される割合を $P(R)$ で示す。(Brink, H., Richards, J. W., Poznanski, D., et al. 2013, "Using Machine Learning for Discovery in Synoptic Survey Imaging"(6))

図 3はテストデータとして用意した約 20,000 天体の Model による予測の分布を示している。予測値 $p(real)$ が 1 に近づくほど Transient 天体である可能性が高くなる。図中赤色の破線は Transient 天体の判定の基準であり、その基準以上であれば Transient 天体として分類される。Artifact と分類された数 $N(B)$ は 19,843 であり、Transient と分類された数 $N(R)$ は 150 であった。この Transient と分類された 150 天体のうち、True Positive である割合 $P(R)$ は、9.28% である。また、False Positive である割合 $P(B)$ を用いて、 $Precision = P(R)/P(B)$ の上限を算出すると 10% であった。これは Model が Transient と判定した天体中 10% は本物の Transient 天体であることを示す。

仮に、PTF で予想される 1 晩で見つかる数である 10^6 天体の検出があったとき、およそ 7,500 天体まで数を削減することができる。またその 7500 天体の中に 750 天体ほど見つかる可能性がある。このように機械学習を用いて多くの Artifact 天体を削減し Transient 天体を検出させることが行われてきた。

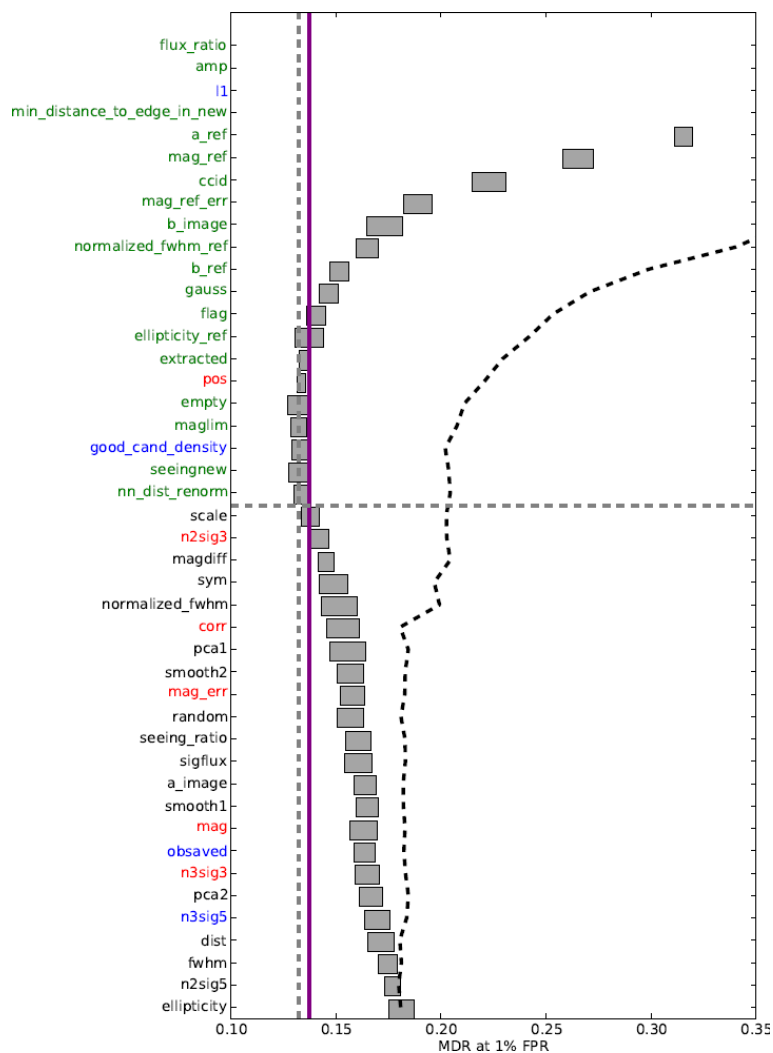


図4 縦軸は特徴量、横軸は $Missed\ Detect_{Rate}$ (MDR) が 1% のときの $False\ Positive\ Rate$ (FPR) を表す。特徴量は、Model の性能に最も寄与しない特徴量が下から順番に並んでいる。グレーのボックスは対応する特徴量のない Model の MDR の平均値 $\pm 1\sigma$ を示す。この試行は 5 回行われ、そのうち 1 回も選択されなかった特徴量は黒色、1 回、2 回、3 回以上選択された場合は、それぞれ赤色、青色、緑色で示す。グレーの垂直の破線は最も良い MDR を示している。グレーの水平の破線よりも上の特徴量を用いた Model の性能は、紫色の実線で示した MDR よりも下回っている。(Brink, H., Richards, J. W., Poznanski, D., et al. 2013, "Using Machine Learning for Discovery in Synoptic Survey Imaging"(6))

2.3 特徴量が Model の性能に与える影響

Random Forest の学習の際には、どの特徴量が Model の分類性能に影響を与えているか分析を行っている。最終的な Model 作成に用いる特徴量には、性能によく影響を与えている特徴量のみを用いることで、すべて用いたときよりも分類性能が高い Model を作成している。

図 4 は下から順に Model の性能に最も寄与しない特徴量を削減していったとき Model の性能がどのように

変化するのかを表している。横軸は $Missed\ Detect\ Rate(MDR) = 1 - TPR$ であり、左ほど Transient 天体を拾う性能が高いことを表す。この試行はランダムに選ばれたデータセットを用いて 5 回行われる。図中のグレーのボックスは対応する特徴量のない Model の $MDR = (1 - TPR)$ の平均値 $\pm 1\sigma$ を示す。縦軸の特徴量について、5 回の試行のうち 1 回も選択されなかった場合は黒色、1 回、2 回、3 回以上選択された場合は、それぞれ赤色、青色、緑色で示す。グレーの垂直の破線は最も良い MDR を表す。紫色の実線はグレーの垂直の破線の $+1\sigma$ であり、この基準を満たした（グレーの水平の破線よりも上の）特徴量は最終的な Model の学習に採用される。

このように Model の性能を高めるために、性能に影響する適切な特徴量を選択することが行われてきた。

2.4 特徴量として画像を用いる

天体から検出された物理量やその天体のカタログ情報のうち、利用できる特徴量を組み合わせて、REAL と BOGUS の違いを捉えられる適切な特徴量を見つけ出し、テストすることは多くの時間を必要とする可能性がある。そこで、撮像された天体の画像のピクセル値を特徴量として与え、Model を作成する試みも行われてきた。

Pan-STARRS はハワイ大学の天文学研究所により、開発および運用されている広視野の観測システムである。Pan-STARRS1 (PS1) では、1.8 メートルの望遠鏡と CCD カメラを用いて全天の観測が行われた。

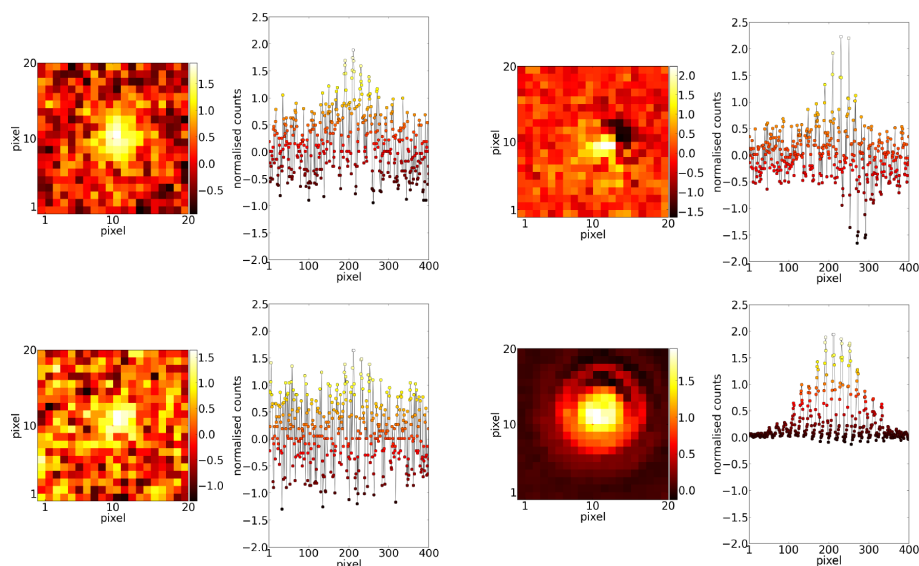


図5 Pan-STARRS1 の REAL-BOGUS 分類で用いられた特徴量を示す。特徴量として画像の value を用いており、その数は 20×20 である。Model に入力する際は 1 次元のデータに整形する。(Wright, D. E., Smartt, S. J., Smith, K. W., et al. 2015, "Machine learning for transient discovery in Pan-STARRS1 difference imaging."(18))

このサーベイで観測された天体の差分画像を特徴量として用いた Random Forest によって分類 Model の作成が試みられた。図 5 は天体の差分画像の例である。左列は Transient の画像であり、右列は Artifact の画像である。ただし、 20×20 の画像を 1 次元のデータとして扱うため、画像の持つ空間的情報が失われる可能性がある。Model の学習には REAL として Transient 天体を 4,800、BOGUS として Artifact 天体を 19,271 の

数を用いた。学習の結果、Model の性能は (TPR, FPR)=(89.4%,1%) であった。

2.5 Random Forest と CNN の分類性能の比較

CNN は画像の空間的情報を捉え特徴量として用いる。それには人による介入がなく、Model がデータから直接特徴量を見つけ出す。The SkyMapper Transient Survey は、Siding Spring 天文台にある SkyMapper 1.3m 望遠鏡を用いた Transient サーベイである。このサーベイの観測データを用いて CNN と Random Forest の Model における分類性能の比較が行われた。Random Forest に与えられた特徴量は画像から抽出された特徴量である。対して CNN では天体の画像を入力とした。このとき天体の画像は、参照画像、ターゲット画像（新規画像）、差分画像の 3 チャンネルとした。特徴量は CNN の Model がこの画像から抽出する。両 Model の学習には REAL として Transient 天体を 223、BOGUS として Artifact 天体を 1,939 の数を用いた。

Model	AUC
Random forest	0.9907
Net1(32,64)	0.9914
Net3	0.9972
E2	0.9946

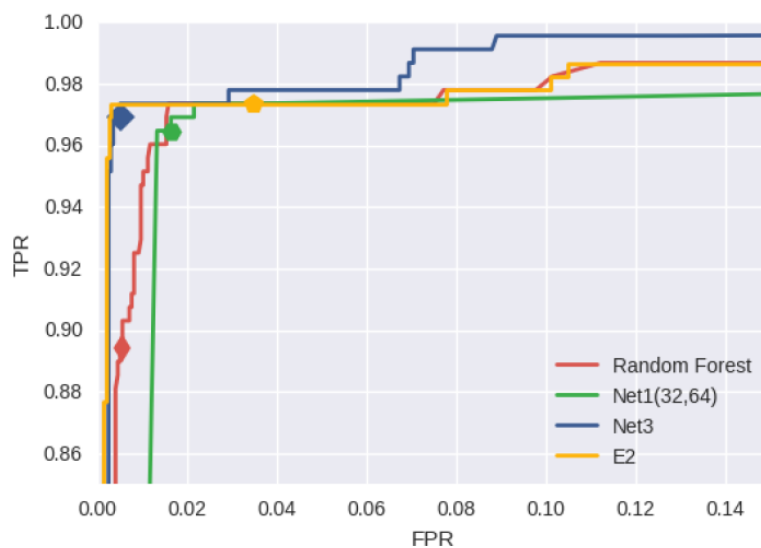


図6 The SkyMapper Transient Survey の観測データを用いて学習した Model の性能を表す。下の図は、横軸に FPR 、縦軸に TPR をとっており分類性能を示す曲線がグラフ左上に近づくほど性能が良い。上の表は、各 Model の下図における曲線下の面積 (AUC) を表す。 AUC が大きいほどグラフが左上に近づくことになり、性能が良い。(Gieseke F., et al. 2017, "Convolutional Neural Networks for Transient Candidate Vetting in Large-Scale Surveys"(10))

図 6 は学習した Model をテストデータで評価した結果を表す。Random Forest 1 種類と CNN 2 種類と Rnandom Forest と CNN のアンサンブルが 1 種類の計 4 種類の Model を比較する。下部は横軸に FPR 、縦

軸に TPR をとったグラフであり、Model の性能を示す曲線が左上に寄るほど性能が高い。なお、この曲線の下側の面積が Area Under the Curve (AUC) であり、1 が最大であり、1 に近づくほど性能が良い。上部は各 Model のその Area Under the Curve (AUC) を示している。2 種類の CNN の Model について、Net3 は Net1(32,64) に比べそれぞれ画像から特徴量を抽出する層が深い。Net3 の Model は Net1(32,64) に比べて画像の形状から特徴量を抽出する機能が豊かであるため性能に差異が生まれたと考えられる。

CNN の Model の性能は Random Forest に匹敵するほどまたは越えるほど良い。CNN の Model は自身が与えられた画像から特徴量を抽出するため、人の手で特徴量を用意するために多くの時間を費やす必要がない。

そこで、本稿では Transient サーベイの REAL-BOGUS 分類に CNN を用いた結果について述べる。

3 Tomo-e Supernova Survey

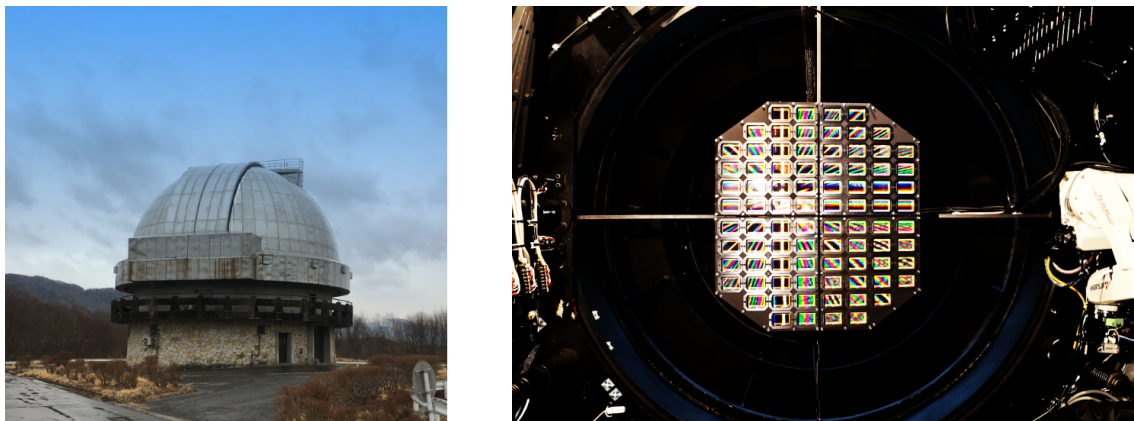


図7 左図：木曾観測所の観測施設のドーム外観。右図：105cm シュミット望遠鏡内の Tomo-e Gozen Camera。 (Shigeyuki Sako's Home, <http://www.ioa.s.u-tokyo.ac.jp/sako/index.html>)

Field of view	9 deg in diameter
Sky coverage	20 deg ²
Frame rate	2 frames/sec (max)
Pixel size	19 microns/pix
Pixel scale	1.198 arcsec/pix
Sensor type	Frontside illuminated CMOS image sensor with microlense array
Sensor format	35 mm full HD, 1k x 2k pix ² , 84 chips
Sensor readout	Rolling readout, Partial readout available
Filter	SDSS-g, Johnson-Cousins I

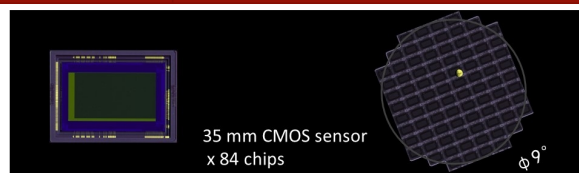


図8 Tomo-e Gozen カメラの性能。84 枚の CMOS カメラ全て合わせて 20 deg² の領域を見ることができる。(the Tomo-e Gozen Project, <http://www.ioa.s.u-tokyo.ac.jp/tomoe/index.html> (22))

東京大学木曾観測所は、理学部附属天文学教育研究センターの観測施設であり、1974 年に開設した。1989 年までは写真乾板、それ以降 CCD センサを用いた観測を行っていた。2018 年からは CMOS センサを用いた観測を行っている。CMOS センサは CCD カメラと比べて高速にデータを読み出せることが特長であり、最短で 2Hz で画像を撮ることが可能である。

木曾観測所の主要観測装置は 105cm シュミット望遠鏡である。図 7 の右図がその望遠鏡の筐体内部の CMOS センサの画像である。現在、この装置に 84 枚の CMOS センサ (合計視野 20 平方度) を搭載し、可視光の広視野サーベイのプロジェクト「Tomo-e Gozen」が進行している。

Tomo-e Gozen では、毎晩数時間おきに夜空を巡回し観測する Tomo-e Supernova Survey が進行している。前章で述べたように重力崩壊型超新星のショックブレイクアウトの観測は、超新星の爆発のメカニズムの解明につながる。Tomo-e Supernova Survey では、継続時間が数分から数時間のショックブレイクアウトを観測し超新星爆発の瞬間を捉えることを目標の一つとし探査を行っている。

3.1 天体の検出

この節では、観測データの処理の過程及びそこから検出された Transient 天体と差分に失敗した天体 (Artifact 天体) とはどのようなものを指すのか、そしてそれらをどのようにして分類するかについて述べる。

3.1.1 観測データの処理の流れ

Tomo-e Gozen サーベイでは、一夜に $\sim 10^5$ もの天体が検出される。そのうち見つかる Transient 天体の数は一夜に ~ 10 天体ほどと見積もられている。それ以外は差分処理に失敗した天体である。Tomo-e Gozen では、いかにこの差分処理に失敗した天体を取り除くことが重要である。本研究では、この検出された天体の中から人の目による超新星の迅速な発見をサポートするために、画像認識の一手法である畳み込みニューラルネットワークによって超新星の候補を選出するシステムを開発した。

1	観測データ (New 画像) に写るすべての天体を検出し測光、測定
2	New 画像と Reference 画像の差分処理
3	差分処理で得られた Subtracted 画像に写るすべての天体を検出し測光、測定
4	Subtracted 画像に写る天体について小惑星、星の検索と、REAL-BOGUS 分類に関する処理
5	Subtracted 画像の限界等級を算出
6	Subtracted 画像から検出された Transient 候補天体について独自の測光処理

表2 観測されたデータの処理のフローチャート

観測されたデータの処理の流れを表 2 に示す。

1. 観測されたデータ (New 画像) はまず、その画像に写るすべての天体を検出し測光、測定を行う。天体の検出および測光、測定には Source Extractor を用いる。検出された天体の像がどれだけ歪んでいるか、どれぐらいの明るさかなどを計測する。
2. Tomo-e で観測された画像 (New 画像) には、天球面上の位置が記録されている。そこで、その New 画像の空の位置周辺の Reference 画像を取得し、その Reference 画像と New 画像の差分をとる。なお、Reference 画像は Pan-STARRS によって観測された画像を用いる。ここで用いる Pan-STARRS 画像は Supernova Survey における Tomo-e 画像よりも天体が明瞭に写っている (SNR が良い) ので、Tomo-e 画像同士で差分を行うよりも、差分処理におけるノイズの影響を小さく抑えることができる。差分処理には hotpants を用いる。Transient サーベイでは、近年 ZOGY という差分処理のソフトウェア^{*1}が用いられることもある。New 画像と Reference 画像はピクセルスケールが異なるため wcsremap

^{*1} Zackay, B., Ofek, E.O., Gal-Yam, A. 2016, "Proper Image Subtraction: Optimal Transient Detection, Photometry, and Hypothesis Testing."(19)

でピンニングと位置合わせを行い New 画像側にピクセルスケールを合わせてから差分処理を行う。

3. 差分処理で得られた画像はまず、その画像に写るすべての天体を検出し測光、測定を行う。天体の検出および測光、測定には Source Extractor を用いる。検出された天体の像がどれだけ歪んでいるか、どれぐらいの明るさかなどを計測する。
4. Subtracted 画像から検出された天体について、まず小惑星の位置情報データベースと照合を行う。照合には astcheck というソフトウェアを用いる。次に、検出された天体は初期処理と画像認識によって、Transient の候補天体か Artifact 天体に分類される。astcheck での照合の結果小惑星のデータと一致した天体も Transient の候補天体から外れる。画像認識の処理の際、Subtracted 画像で検出された天体の位置を基準に 29×29 pixel のサイズに天体の画像を切り出す。画像は Reference 画像、New 画像、Subtracted 画像からそれぞれ切り出され、すべて画像認識の処理に用いられる。初期処理と画像認識の処理は後述する。

Tomo-e Supernova Survey では、異なる観測データでもう一度上記の処理を経て Transient の候補天体として検出された場合に Transient 天体として Tomo-e Transient Server に登録される。Tomo-e Transient Server は、今までに検出された Transient 天体についての情報を確認できる Web Server であり、登録された Transient 天体が、Artifact 天体か”本物の Transient 天体”かどうか人の目によるチェックが行われる。この Web Server 上では光度曲線、Transient 天体として登録される以前の cutout 画像などが確認でき、これらの情報から本物の Transient 天体かどうかの判断を下す。

3.1.2 Transient 天体と Artifact 天体

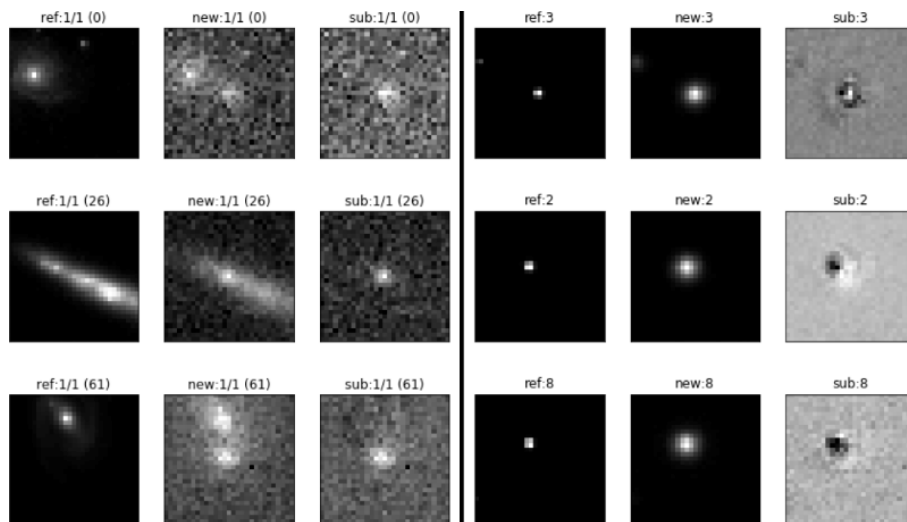


図9 Tomo-e で撮影された天体の例。中央の黒線を挟んで左側が Transient 天体、右側が Artifact 天体を表す。それぞれ左の列から、Reference 画像、New 画像、Subtracted 画像を表す。

図9は Tomo-e で撮影された天体のうち Transient であると判明している天体（左3列）と Artifact 天体

(右3列)の例である。右3列と左3列はそれぞれ、左から Reference 画像、New 画像、Subtracted 画像である。Transient 天体は、New 画像では母銀河と共に写ることが多い*²。Artifact 天体は、上段のように中心付近が引き残りその近傍で sky value*³よりも落ち込む PSF を持つもの、中段や下段のような双極子的 PSF を持つものなどが存在する。これらの天体を画像認識によって Transient 天体と Artifact 天体に分類する。

3.2 天体の検出における初期処理について

前節のデータ処理4項目の画像認識を行う前の初期処理について詳しく述べる。初期処理では主に以下の3つの処理を行う。

1. Mask 領域から検出された天体を除外する。

Reference 画像において、New 画像に対応する領域が取得できない場合が存在する。その Subtracted 画像において、Reference 画像が存在しない領域 (Mask 領域) から検出された天体は、差分処理を正常に完了できない。そのため、それらの天体を Transient 候補から除外する。

2. 画像端に写る天体を除外する。

hotpants では、画像引き算の際、Subtracted 画像の端が Mask される。そのため、その Mask 領域に干渉する天体を Transient 候補から除外する。

3. Reference 画像に写る飽和した天体を除外する。

Reference 画像に写る天体のうち、飽和した天体はその中心部分が Mask される。図10のように Mask された飽和天体は差分処理を行うと Subtracted 画像で引き残ってしまう。そのため、その Mask された飽和天体を Transient 候補から除外する。

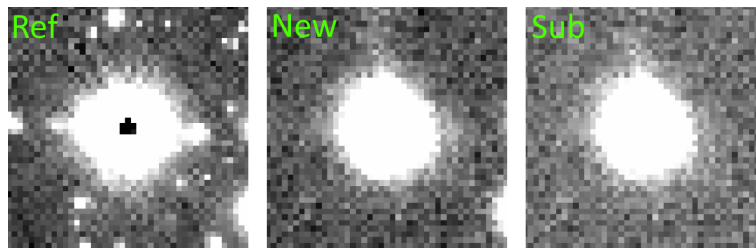


図10 Reference 画像で飽和し Mask された天体の差分処理前後の画像を表す。左から Reference 画像、New 画像、Subtracted 画像。

*² 地球から母銀河までの距離が遠い場合、超新星は銀河全体の光度に匹敵するほど明るいので、母銀河が写らず超新星のみが写る可能性もある。

*³ sky value とは天体が写っていない pixel がもつ値のこと。大気からの微弱な光を検出している。

3.3 画像認識による天体の検出

画像認識による天体の分類を行うためには、まず既知の Transient 天体と Artifact 天体を用いて分類を行う Model の学習を行う。画像認識には畳み込みニューラルネットワークを用いる。

畳み込みニューラルネットワークとは、主に畳み込み層とプーリング層、全結合層から成る。

3.3.1 全結合層

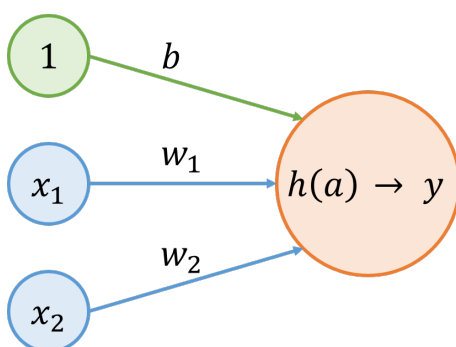


図11 全結合層のニューラルネットワークの基本構成の模式図。入力信号が左の円で示したニューロンに入力され、右の円で示したニューロンに信号を伝える。信号を伝える際、右側の各ニューロンで重みを掛けてから足し合わせる。その足し合わせた信号を活性化関数を用いて変換し、信号を出力する。

全結合層のニューラルネットワークの基本的な構成を図 11 に示す。図の○で表現されたものをニューロンと呼び、第 0 列目を入力層、第 1 列目を出力層という。

$$a = b + w_1x_1 + w_2x_2 \quad (1)$$

$$y = h(a) \quad (2)$$

入力信号のうち緑色で表されたニューロンはバイアスでありその重みは b と表している。 w は各 x の重みを表す。入力層 x_1, x_2 は、入力される信号を表している。式 1、2 に示すように、入力信号に w_1, w_2 の重みをかけ、足し合わせた a を活性化関数 h により変換し出力層の信号 y を出力する。

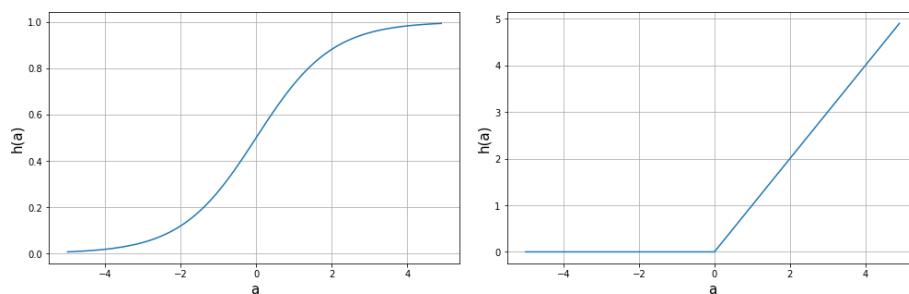


図12 活性化関数の例。左側が Sigmoid 関数を表し、右側が Rectified Linear Unit (ReLU) 関数を表す。

$$h(a) = \frac{1}{1 + \exp(-a)} \quad (3)$$

$$h(a) = \begin{cases} a & (a > 0) \\ 0 & (a \leq 0) \end{cases} \quad (4)$$

活性化関数はニューラルネットワークの非線形応答を実現する重要な関数であり、様々な関数がいられる。そのうち、Model を作成する上で用いる活性化関数を紹介する。図 12に今回用いた 2 つの活性化関数を表す。左図は、Sigmoid 関数と呼ばれ、式 3で表される。Sigmoid 関数は入力された信号 a に対して 0 から 1 の範囲内で出力する。今回この関数は後述するニューラルネットワークの出力層で用いており、Transient 天体かどうかを確率として表す。右図は、Rectified Linear Unit (ReLU) 関数と呼ばれ、式 4で表される。この関数は 0 を基準とし基準以下なら 0 を、基準以上ならば入力 a をそのまま出力する。

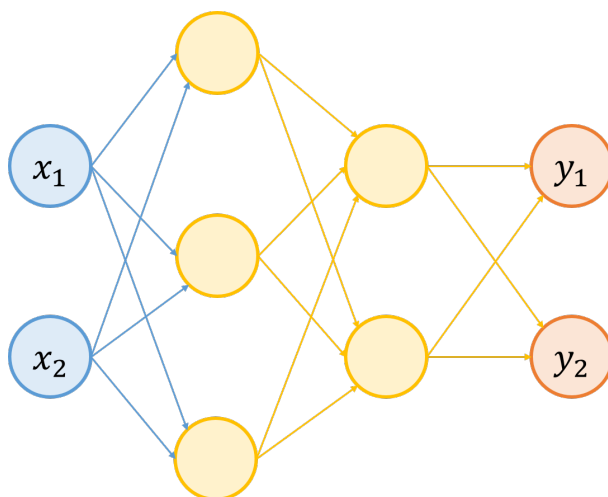


図13 今回作成した Model の全結合層の層数と同じ 3 層のニューラルネットワークを示す。ただし Model には図で示したニューロンよりも多くのニューロンが接続されている。

今紹介したニューラルネットワークは 1 層^{*4}のニューラルネットワークである。さて、図 13に今回作成した Model の全結合層の層数と同じニューラルネットワークを示す。作成した Model は図 13で表されている

*4 出力層を除いた層の数

より多くのニューロンが接続されているが、単純化のため少数のニューロンを用いて説明する。

青色で示したニューロンの層は入力層を表し、赤色で示したニューロンの層は出力層である。黄色で示したニューロンの層は隠れ層と呼ぶ。なお、ここではバイアスを表すニューロンは省略している。

$$\begin{aligned}
 \mathbf{A}^{(1)} &= \mathbf{X} \mathbf{W}^{(1)} + \mathbf{B}^{(1)} \\
 \mathbf{A}^{(1)\text{T}} &= (a_1^{(1)}, a_2^{(1)}, a_3^{(1)}) \\
 \mathbf{X}^{(1)\text{T}} &= (x_1^{(1)}, x_2^{(1)}) \\
 \mathbf{B}^{(1)\text{T}} &= (b_1^{(1)}, b_2^{(1)}, b_3^{(1)}) \\
 \mathbf{W}^{(1)} &= \begin{pmatrix} w_{11}^{(1)} & w_{21}^{(1)} & w_{31}^{(1)} \\ w_{12}^{(1)} & w_{22}^{(1)} & w_{32}^{(1)} \end{pmatrix} \tag{5}
 \end{aligned}$$

第1層目の重み付き和を式5に示す。各変数の肩⁽¹⁾は第1層目を表す。各変数の右側の添え字はどのニューロンかを表し、左側の添え字は次層の接続先のニューロンを表す。入力 $\mathbf{X}^{(1)}$ に重み $\mathbf{W}^{(1)}$ を掛けバイアス $\mathbf{B}^{(1)}$ との和をとり、重み付き和 $\mathbf{A}^{(1)}$ を求める。さらに $\mathbf{A}^{(1)}$ を活性化関数で変換し次の第2層の隠れ層の入力とする。第2層以降も同様に第1層のニューロンの信号から次の信号の計算を行う。今回作成した Model の活性化関数は、隠れ層に ReLU 関数、出力層に Sigmoid 関数を用いる。

3.3.2 畳み込み層とプーリング層

これまで述べてきた全結合層は1次元のデータの入力を要求する。すなわち画像のような2次元、3次元のデータは1次元に整形し入力する必要がある。これでは、画像に含まれる空間的情報を活かして画像の形状に潜む本質的なパターンの学習を行うことが難しい。

そこでこの問題を解決するために畳み込み層とプーリング層を導入する。畳み込み層とプーリング層では画像の形状のパターンを抽出することができ、抽出した特徴量を用いることにより画像に含まれる空間的情報を学習することができる。この畳み込み層とプーリング層、全結合層から成る構成を畳み込みニューラルネットワークと呼ぶ。

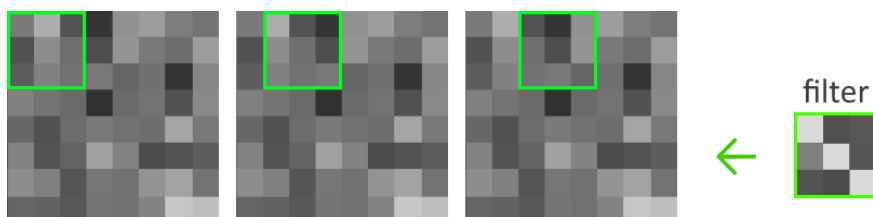


図14 8×8の画像の畳み込み演算の模式図。畳み込む filter のサイズは3×3である。ウィンドウ内のそれぞれの場所で画像の value と filter の value を掛けた値をすべてを足し合わせ、1つの値を算出する。この処理を緑色で示した3×3の filter のウィンドウを一定の間隔でスライドさせて画像に適用させる。その各結果を6×6の画像として出力する。

畳み込み層では、画像のような2次元、3次元のデータに対して畳み込み演算を行うことで画像から特徴を抽出する。畳み込み演算では、入力された画像に対して任意のフィルターを適用する。図14は畳み込み演算の例である。緑色で示した3×3の filter のウィンドウを一定の間隔でスライドさせて8×8の画像に filter を適

用していく。ウィンドウ内のそれぞれの場所で画像の value と filter の value を掛けた値の 9 つすべてを足し合わせる。この処理をスライド毎に行うと画像の value と filter の value を掛けた値で構成された 6×6 の画像を得る。この画像の value は、filter の持つ形状に合致している場所では値は大きくなり、合致していないと値は小さくなる。すなわち、画像のもつ形状を filter によって抽出している。

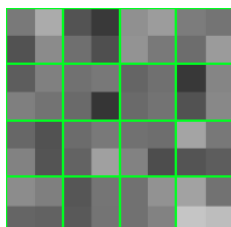


図15 プーリング処理の例。図は 2×2 の領域を一つの value として集約する。ここでは集約の方法として Max pooling を用いる。

プーリング層は、画像の大きさを小さくする処理を行う。図 15 は緑色で示した各 2×2 の領域をひとつの value として集約する処理を示している。集約の方法はいくつか存在し、ここでは Max pooling について述べる。Max pooling は領域内の最大値をとる方法である。図 15 の場合、各 2×2 の領域内の最大値をとった 4×4 の画像が出力される。プーリング層は画像を小さくすることで、学習するパラメータ数の削減に寄与するだけでなく、より大域的な特徴の抽出に貢献する。例えば、プーリング処理後に畳み込み処理を行うことで、プーリング処理前と同じサイズのフィルターで、より広範囲の画像の特徴を抽出することが可能となる。

3.3.3 学習

畳み込みニューラルネットワークにおける学習とは、ニューラルネットワークの重み w とバイアス b 、畳み込み層における filter の各パラメータを訓練データに適応するように調整することである。学習にはミニバッチ学習という方法を用いる。具体的な手順は、まず訓練データを N 分割する。 N 分割されたデータをミニバッチという。その後、以下の手順を行う。

1. ミニバッチを決定する。
訓練データからランダムにミニバッチを選び出す。
2. ミニバッチの損失関数の値を減らす勾配方向を計算する。
ミニバッチに定義された損失関数の値を減らすために各重みパラメータの勾配を決定する。勾配は損失関数の値を最も減らす方向を示す。
3. 重みパラメータを勾配方向に微小量だけ更新する。

以上の手順を N 回繰り返す。そして N 回学習を終え、このミニバッチ学習ですべての訓練データについて学習した段階を 1 epoch という。ここで損失関数とは、訓練データの正解ラベルと Model の予測ラベルのずれを表す関数である。この損失関数の値が小さくなるほどより正確な Model となる。

$$E = -\frac{1}{N} \sum_i^N E_i \quad (6)$$

各 epoch の損失関数 E は、式 6 から求める。すなわち各 epoch の損失関数 E は各ミニバッチの損失関数 E_i の平均とする。Model の学習は、1epoch 目の損失関数 E を計算した後もう一度訓練データを N 分割し、上記の手順を繰り返して損失関数の減少が十分収束するまで epoch を回す。このようにして Model の学習が完了する。

$$E_i = -\sum_j^n (d_j \log y_j + (1 - d_j) \log(1 - y_j)) \quad (7)$$

本研究では損失関数として、二値交差エントロピーである式 7 を採用する。 E_i は i 番目のミニバッチにおける損失関数を表す。また、 N 分割されたミニバッチのデータサイズを n としている。 d_j は、ミニバッチ内の j 番目のデータの正解ラベルを表す。正解ラベルは REAL ならば 1、BOGUS ならば 0 をとる。 y_j は、その j 番目のデータに対する Model の予測値であり 0 ~ 1 の範囲の実数をとる。1 に近いほど REAL である確率が高い。右辺について、 j 番目の正解ラベルが REAL のとき、第二項は 0 となるため、第一項によって損失が算出される。正解ラベルが BOGUS の場合は逆に第二項によって損失が算出される。右辺 $\log y_j$ 、 $\log(1 - y_j)$ の値は、REAL と BOGUS に対する予測値が正解ラベルの値に近いほどそれぞれ 0 に近づき、そのデータでの損失が小さくなる。すなわち Model の学習とは、各ミニバッチの損失関数 E_i を小さくすることに他ならない。

3.3.4 Model の構成

Layer (type)	Output Shape	Param #
conv2d_1 (Conv2D)	(None, 25, 25, 32)	2432
max_pooling2d_1 (MaxPooling2D)	(None, 12, 12, 32)	0
conv2d_2 (Conv2D)	(None, 10, 10, 64)	18496
max_pooling2d_2 (MaxPooling2D)	(None, 5, 5, 64)	0
dropout_1 (Dropout)	(None, 5, 5, 64)	0
flatten_1 (Flatten)	(None, 1600)	0
dense_1 (Dense)	(None, 128)	204928
dense_2 (Dense)	(None, 64)	8256
dense_3 (Dense)	(None, 1)	65
Total params: 234,177		
Trainable params: 234,177		
Non-trainable params: 0		

図16 今回作成した Model の層構成を示す。左の列から層の名称、各層が出力するデータのサイズ、学習を行うパラメータの数を表す。

図 16 に今回作成した CNN Model の構成を示す。1 列目は用いた層を記している。2 列目はその層が出力するデータのサイズを表す。3 列目は学習するパラメータ数を示している。主な構成は、畳み込み層 (conv2d) と

それに続くプーリング層 (max_pooling2d) がそれぞれ 2 層あり、全結合層 (dense) が 3 層のニューラルネットワークである。入力データは、Reference 画像、New 画像、Subtracted 画像の 3 チャンネルを用いる。まず入力データは、データのサイズが $29 \times 29 \times 3$ であり 1 層目の畳み込み層に入力される。1 層目の filter のサイズは 5×5 であり、3 チャンネルそれぞれにはランダムに初期化された 32 枚の filter で構成される。また、32 枚それぞれの filter には 3 チャンネル共通の 1 つのバイアスパラメータが用いられる。よって 1 層目の学習パラメータ数は、 $5 \times 5 \times 32 \times 3 + 32 = 2432$ である。また入力された画像は 32 枚の filter の畳み込み処理によって $25 \times 25 \times 32$ のサイズで 3 チャンネル出力される。2 層目は Max Pooling 層であり、入力画像のサイズを縮小する。これにより各チャンネル $12 \times 12 \times 32$ のサイズとなる。3 層目は 2 つ目の畳み込み層である。3 層目の filter のサイズは 3×3 であり、64 枚の filter で構成される。その後データは dropout 処理と flatten 処理を経て全結合層へと流れる。dropout とは、一定の確率でランダムに選び出した重みの更新を止める処理である。dropout は、それによって学習の計算コストが増大するが、訓練データに対する過学習を抑制することができる。flatten とは、入力のデータを 1 次元に変換する処理である。1 つ目の全結合層は入力数 1600 出力数 128 で、活性化関数に ReLU 関数を用いる。ニューロン同士の重みのパラメータ数はバイアスを含めて $1600 \times 28 + 128 = 204928$ である。2 つ目の全結合層は入力数 128 出力数 64 であり、活性化関数は 1 つ目と同様 ReLU 関数を用いる。最後の全結合層は出力層であり、活性化関数に Sigmoid 関数を用いることで出力値を 0 ~ 1 の範囲としその Model の予測値を確率として扱う。

本研究では Model の作成およびパイプラインの処理のために Python を用いる。Model の構築、学習には Keras というモジュールを用いる。今回用いた Keras ではバックエンドのモジュールとして tensorflow を選択した*⁵。

また、Model は、Tomo-e Gozen の CMOS センサ 84 枚それぞれに対応した Model を作成する。

*⁵ Python: ver.2.7.15, Keras: ver.2.1.6, tensorflow: ver.1.8.0

4 学習データ

この章では、実際に学習データの作成方法について述べる。一般的に畳み込みニューラルネットワークの学習には数千から数万のデータを用意する必要がある。本研究では、REAL:14,000、BOGUS:14,000 の数の学習データを用意し二値分類を行う。

4.1 REAL: Artificial Real

REAL の学習データとしては、画像内に映る恒星の PSF から人工的に天体を作成する。「Tomo-e Gozen」で得られる Transient 天体の数は一夜に ~ 10 天体ほどと見積もられており、十分な観測データを用意することができない。そこで学習データの作成には、観測された恒星の画像から得た統計的な光の拡がりを用い、人工的な Transient(Artificial Real) のデータを作成し学習データとした。

4.1.1 Artificial Real の PSF

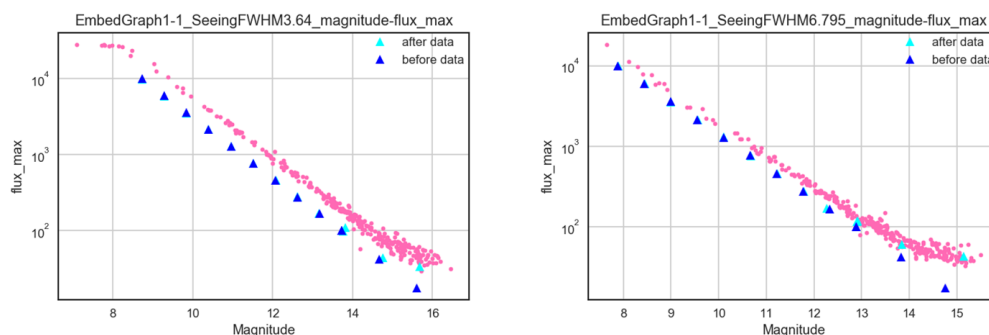


図17 PSF としてガウシアンを用いて作成した Artificial Real とその Fits 画像内に写る恒星の *Magnitude* の比較。右図は *Seeing FWHM* = 3.64 の画像、左図は *Seeing FWHM* = 6.795 の画像から検出した恒星のパラメータを表す。青色の点と水色の点が Artificial Real を示す。そのうち、青色の点は New 画像に埋め込む際のパラメータを示し、水色の点は Source Extractor を用いて New 画像から検出したパラメータを示す。

人工的な Transient(Artificial Real) 天体を作るにあたって、PSF を決定する必要がある。星は地球から見てはるか遠方にあるため、地球から見たとき点源に見えるはずだが、実際には地球大気との相互作用により光が散乱され、像がぼやける。点像からぼやけた像を表す関数を PSF(Point Spread Function) という。

図 17は、PSF としてガウシアンを選択し Artificial Real を作り、実際に写る天体と *Magnitude* と *Flux max* を比較した図である。

$$Magnitude = Magnitude_{zero} - 2.5 \times \log_{10} flux \quad (8)$$

Magnitude は、式 8 で表される。*Magnitude_{zero}* は各 pixel の 1 カウントが何等に対応するかを決定する。*Flux max* はその天体の像のうち最も pixel value が大きい値である。図中青い点は、New 画像に埋め込む際に与えた *Flux max* とその *Flux max* から算出された *Magnitude* を示している。水色の点は、埋め込んだ Artificial Real の Source Extractor の検出から得られる *Flux max* と *Magnitude* を示している。実際に写

る天体の Source Extractor の検出から得られた *Flux max* と *Magnitude* は桃色で示す。実際に写る天体と Artificial Real を比べると、同じ *Flux max* でも Artificial Real の方が実際に写る天体よりも *Magnitude* が大きい。すなわち、Artificial Real の方が実際の天体よりも明るく検出されている。

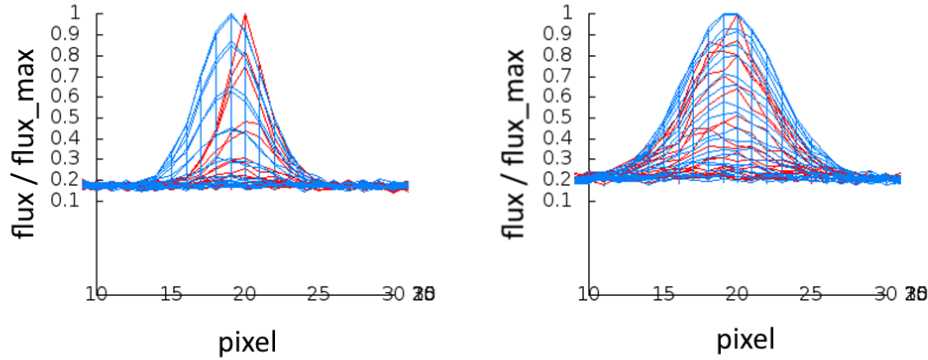


図18 画像に写る恒星と作成した Artificial Real の像の比較。青色の曲面が Artificial Real、赤色の曲面が実際の恒星の像を示す。右図は *Seeing FWHM* = 3.64、左図は *Seeing FWHM* = 6.795 での恒星と Artificial Real の像を表す。Artificial Real の像の方が実際に写る恒星の像よりも太っていることがわかる。

図 18は、pixel の平面にその pixel での value を縦軸にとっている。この value はその天体の最大値が 1 となるよう *Flux max* で規格化している。左図は *Seeing FWHM* = 3.64 の実際の天体と Artificial Real の像を、右図は *Seeing FWHM* = 6.80 の実際の天体と Artificial Real の像を比較している。

図 18から分かるように実際の天体の像は PSF としてガウシアンを採用した Artificial Real よりも痩せている。図 17の Artificial Real が実際の天体よりも明るい原因はこの像の違いによるものと考えられる。*Magnitude* は開口内の *flux* の積分から算出されるので、同じ *Flux max* であっても Artificial Real よりも痩せている実際の天体の像の方が *flux* の積分値は小さい。そのため図 17の Artificial Real よりも本物の天体の方が *Magnitude* 大きくなった（暗くなった）ことが窺える。

そこで、画像内に写るすべての実際の星の像の Median から Artificial Real の PSF を作成する。

$$f_{\text{Artificial real}}(x, y) = \text{Median} \left(\frac{f_i(x, y) - \text{Sky value}_i}{f_i(x_{\text{max}}, y_{\text{max}}) - \text{Sky value}_i} \right) \quad (9)$$

Artificial Real の psf の導出を、式 9に示す。 i は、同じ fits 画像内に映る星に対する通し番号である。 $f_i(x, y)$ は円形の開口内に対して適用され、その開口内の任意の x, y における *pixel* の value を表す。この開口半径 r_{aperture} は各星の *Flux max* からの *pixel* の距離であり、全ての星に対して $r_{\text{aperture}} = 14$ で一定とした。

$x_{\text{max}}, y_{\text{max}}$ は開口半径内でそれぞれの軸方向で最も大きな value を持つ x と y を指す。 Sky value_i は各星の開口の外側の value の Median をとった値である。開口の外側の具体的な範囲については図 19に示す。

各星は 29×29 のサイズで切り出される。次にこの cutout 画像の開口半径外の領域から *skyvalue* が算出される。そして各星の開口内の像から算出された *sky value* を引き、その value の最大値が 1 となるように像の規格化を行う。それらの処理を施した fits 画像内の全ての星の Median をとったものを Artificial Real の

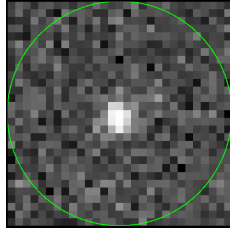


図19 Artificial Real 作成の際に、new 画像の実際の星の像を緑色の開口内から測る。星の画像は 29×29 のサイズで切り出される。Sky value は、この緑色の開口外の四隅の領域のピクセルから Median で算出している。

PSF としている。

この PSF を用いて $flux\ max = 310, 1000, 3100, 10000$ のピーク値をそれぞれ持つ Artificial Real を、seeing が異なる 2 つの fits 画像で作成した。

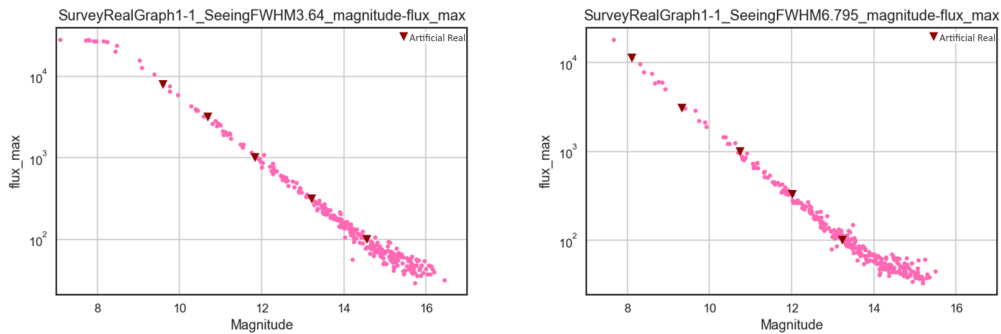


図20 実際の天体から得られた PSF を用いて作成した Artificial Real とその Fits 画像内に写る恒星の Magnitude の比較

図 20は、画像内に埋め込んで Source Extractor で検出した Artificial Real と、その画像から検出された天体の $Flux\ max$ と $Magnitude$ を表す。図中の濃い赤色の点は検出された Artificial Real、桃色の点は画像から検出された天体である。PSF としてガウシアンを選択した Artificial Real では、実際の天体よりも明るく検出されていたが、実際の天体から得られた PSF を用いた Artificial Real では、図のように、画像から検出された天体の分布に一致している。

よって、本稿における学習データの作成は、実際の天体から得られた PSF から Artificial Real を作成する。

4.1.2 Artificial Real の埋め込み

超新星は星の進化の果てである。星は銀河に密集しており、超新星爆発は、ほとんど銀河内で発生すると考えると、Artificial Real は銀河の像の周りに埋め込むことが好ましい。また、超新星の明るさは銀河全体の光度に匹敵するため、母銀河は画像に写っていないが、超新星のみ写る可能性がある。このとき、Reference 画像には何も映っていないが New 画像に超新星のみが写っていることになる。そのため、銀河周りに埋め込む Artificial Real に加え画像内にランダムに Artificial Real を埋め込む。

埋め込む Artificial Real の明るさは 14 ~ 22 等級とした。学習データ作成に用いたデータは 2019 年 9 月

25日-10月25日に観測された画像である。前述のとおり、学習データのうち REAL ラベルは2種類のデータを用いる。1つ目のデータは、画像内から検出された星から作られた Artificial Real を同画像内にランダムに埋め込んだデータで、7,000 天体作成する。2つ目のデータは画像内から検出された天体のうち、光源が拡がっている天体周りに埋め込んだデータで、こちらも7,000 天体作成する。光源が拡がっている天体は、画像内から検出された天体のうち Pan-STARRS に extended の属性が登録された（光源が拡がっている）天体である。光源が拡がっている天体を母銀河とし、画像内に写るすべての光源が拡がっている天体の周辺に対して天体を埋め込む。

4.2 BOGUS: Artifact

Artifact 天体については、画像内から得られた Artifact 天体を学習データとしている。ただし、この Artifact 天体の学習データには”本物”の transient 天体が紛れ込んでいる可能性がある。この BOGUS の学習データの汚染については、14,000 の Artifact 天体中に紛れ込むであろう未確認の Transient 天体の数の期待値は大きく見積もって、数個以下であり学習に深刻な影響を与えないと結論付けた。

4.3 評価に用いる指標

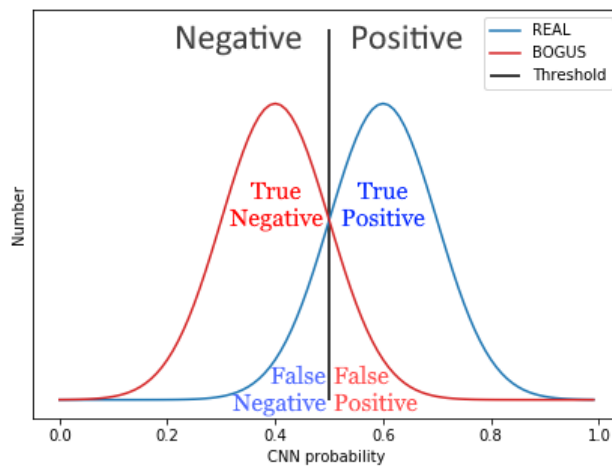


図21 CNN Model が分類した REAL と BOGUS の分布の図。図中 *Threshold* よりも *CNN Probability* が大きければ Positive、すなわち REAL として分類され、小さければ Negative、すなわち BOGUS として分類される。REAL と BOGUS の正しく分類されたデータはそれぞれ True Positive、True Negative と呼ばれ、REAL を誤って BOGUS に分類したデータを False Negative、BOGUS を誤って REAL に分類したデータを False Positive と呼ぶ。

この節では、作成した CNN Model 及び初期処理の性能の評価方法について述べる。REAL と BOGUS が予め判明している天体を CNN Model 及び初期処理によってどちらに分類したかを評価する。

図 21 は、学習した CNN Model があるデータを与えられたときの REAL と BOGUS の *CNN Probability* の分布を示している。*CNN Probability* とは、CNN Model が REAL かどうかを 0 から 1 の範囲で予測

した実数値である。1に近いほど REAL である可能性が高い。図中「*Threshold*」とは、その値よりもデータの *CNN Probability* が小さければ Negative (BOGUS に分類する) とし、大きければ Positive (REAL と分類する) として扱う。REAL の分布について、*Threshold* よりも大きい *CNN Probability* をもったデータは、正しく REAL と分類される。このデータを True Positive と呼ぶ。また、*Threshold* よりも小さい *CNN Probability* をもったデータは、誤って BOGUS と分類される。このデータを False Negative と呼ぶ。BOGUS の分布についても同様に、正しく BOGUS と分類されたデータを True Negative と呼び、誤って REAL と分類されたデータを False Positive と呼ぶ。

$$\text{True Positive Rate} = \frac{\text{True Positive}}{\text{True Positive} + \text{False Negative}} \quad (10)$$

$$\text{False Positive Rate} = \frac{\text{False Positive}}{\text{True Negative} + \text{False Positive}} \quad (11)$$

次に Model の評価に頻繁に用いる指標である *TruePositiveRate* と *FalsePositiveRate* について述べる。*TruePositiveRate* は、式 10 で表されるように REAL 全体のうち CNN Model が REAL と正確に判定したデータの割合である。*FalsePositiveRate* は、式 11 で表されるように BOGUS 全体のうち CNN Model が REAL と誤判定したデータの割合である。

実際に CNN Model を用いる Tomo-e Transient Survey の環境は、REAL が $10^0 \sim 10^1$ /night、BOGUS が $10^5 \sim 10^6$ /night ほど検出されると見積もられており、True Positive よりも遥かに False Positive の数が多くなることが予想される。

5 Artificial Real を用いた評価

この章では REAL として Artificial Real を用いて、作成した CNN Model の性能評価を行う。Model の評価には 84 枚の CMOS センサのうち、中央の Q1 1-1 の検出器に対応する Model を用いる。

5.1 テストデータについて

Model の評価には、REAL、BOGUS ともに学習データと同様の方法で作成した Artificial Real、Artifact からなるデータを用いる。テストデータは学習データに使われていないデータを用いている。テストデータの数は REAL、BOGUS ともに 2000 天体である。

5.2 Model の ROC curve

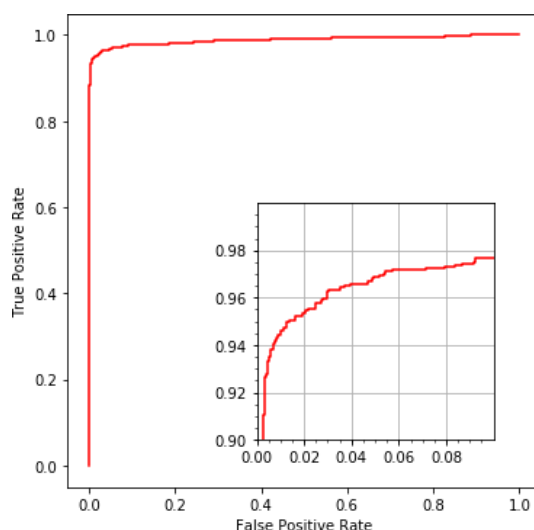


図22 テストデータに対する Model の受信者操作特性曲線 (ROC curve) を表す。ROC curve は *CNN probability threshold* を 0 から 1 まで動かしたときに得られる *False Positive Rate* と *True Positive Rate* の変化を表す。この曲線が左上に近づくほど性能が良い。

図 22はテストデータに対する Model の受信者操作特性曲線 (ROC curve) を表す。横軸は *False Positive Rate(FPR)* である。*FPR* は Model が BOGUS を誤って REAL と分類した割合であり、小さいほど性能が良い。縦軸は *True Positive Rate(TPR)* であり、Model が REAL を正しく REAL と分類した割合を表すため、高いほど性能が良い。この曲線は図 21における *CNN probability threshold* を 0 から 1 まで動かしたときに得られる *FPR* と *TPR* の変化を表す。またこの曲線下の面積 *AUC* は 0.9883 であった。*AUC* は最大値である 1 に近づくほど性能が良いことを示す。

この曲線における *CNN Threshold* = 0.5 での Model の分類性能は、*FPR* = 1.25%、*TPR* = 94.8% であった。この章における以降の評価では *CNN Threshold* = 0.5 のときの性能について述べる。

5.3 SNR について

天体のパラメータの検出は Source Extractor を用いて行う。

$$SNR = \frac{flux}{flux_err} \quad (12)$$

Signal to Noise Ratio (SNR) は式 12 で求められる。 $flux$ は、円形の aperture 内の像をガウシアンでフィッティングした際の積分値である。 $flux_err$ は、各ピクセルの値のフィッティングからの誤差を表す。すなわち、 SNR が大きいほど天体の像が明瞭であることを表す。

5.4 Model の SNR 依存性

この節では天体の差分画像における SNR についての結果と考察を示す。

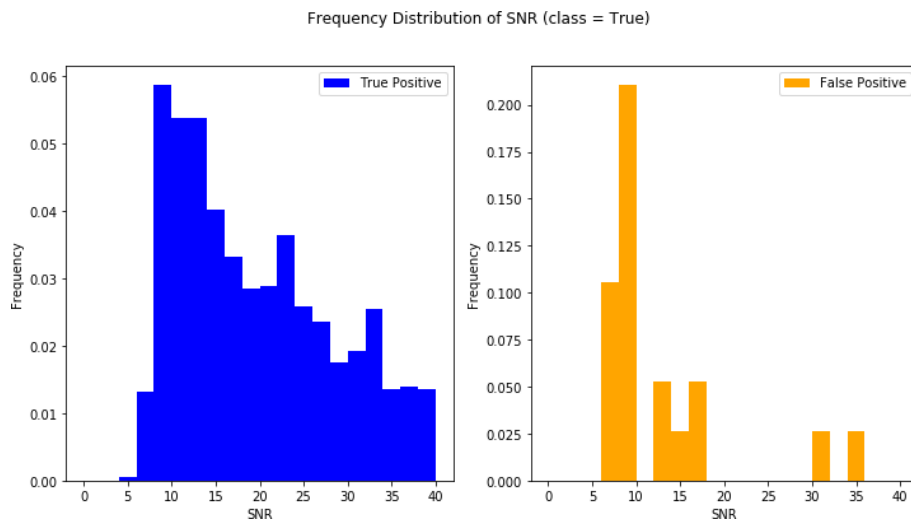


図23 True Positive と False Positive それぞれの SNR の度数分布を表す。

図 23 は Model が REAL に分類した天体のうち、ラベルが REAL である True Positive と、ラベルが BOGUS である False Positive の SNR の度数分布を表す。左図が True Positive の分布であり、右図が False Positive の分布である。False Positive の SNR の分布は True Positive の分布に比べて、高 SNR の数が少ないことが分かる。

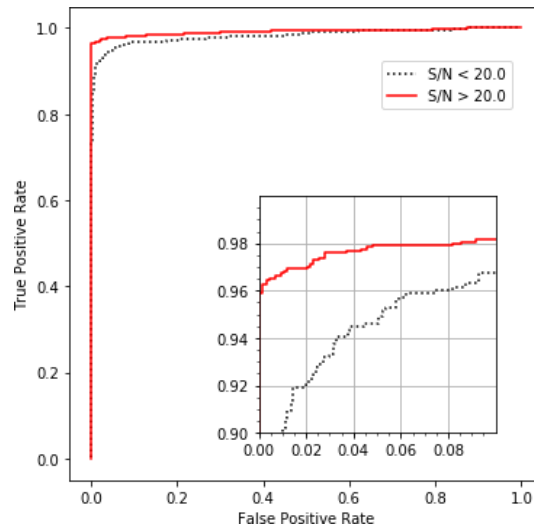


図24 $SNR\ threshold = 20$ で、それ以上の天体の ROC curve (赤色実線) と、それ未満の天体の ROC curve (黒色破線) を示す。 $SNR > 20$ の天体の方が AUC のスコアは高く、 $SNR < 20$ の ROC curve を上回っており、天体の像が明瞭に写っていると Model の分類性能が高いことを表す。

そこで、 $SNR = 20$ でデータを分けて Model の分類性能を評価する。図 24は、 $SNR > 20$ の天体の分類性能 (赤色実線) と $SNR < 20$ の天体の分類性能 (黒色破線) を示す。 $SNR > 20$ のグループの AUC は 0.9918、 $SNR < 20$ のグループの AUC は 0.9818 であり、 $SNR > 20$ のグループの方が SNR は高い。

これは天体の像が明瞭に写っていると Model の分類性能が高いことを表す。

天体の SNR に対して Model の分類性能がどれだけ高くなるかの定量的な評価は次章の Real Transient を用いた評価で行う。

6 Real Transient を用いた評価

この章では Real として Transient Name Server というウェブサーバに報告された本物の Transient を用いて作成した CNN Model の性能評価を行う。

6.1 Real Transient のテストデータ

Model の評価に用いるデータのうち REAL として用いるデータは、前章の Artificial Real で構成されたシミュレーションデータに変わり、実際に発見された突発天体が報告されているウェブサイトより、発見時刻以降に Tomo-e の観測視野内に入っていたデータを用いる。BOGUS として用いるデータは、前章と同様に Model 学習時と同様の方法で作成した Artifact を用いる。

6.1.1 Transient Name Server

Transient Name Server とは、世界中の観測者が発見した超新星やその候補の突発天体が報告されている超新星の発見・命名方法に関する自動システムである。学習させた Model の評価にはこの Transient Name Server (TNS) に報告された突発天体のうち、その天体の発見時刻以降に Tomo-e の観測視野内に入っており、差分画像から天体として検出され、登録されたデータ (Tomo-e source_sub) を用いる。

6.1.2 テストデータ

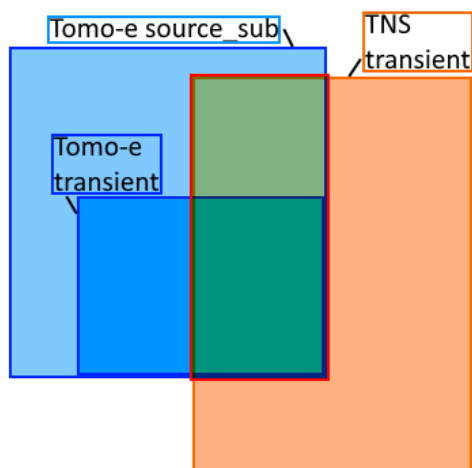


図25 Tomo-e の検出天体と、Transient Name Server (TNS) の天体の関係を示した図。Tomo-e source_sub: Tomo-e の差分画像から検出された天体。TNS transeint: TNS に報告された天体。Tomo-e transient: Tomo-e source_sub の天体のうち突発天体と判定された天体。

図 25 について、Tomo-e transient は Tomo-e source_sub の天体から初期処理と CNN Model の分類の結果突発天体と判定された天体である。Tomo-e transient には突発天体だけでなく Artifact も含まれている。図中赤枠内のデータは、TNS に登録されている天体のうち、発見時刻以降に該当の座標から 3 arcsec(Tomo-e の

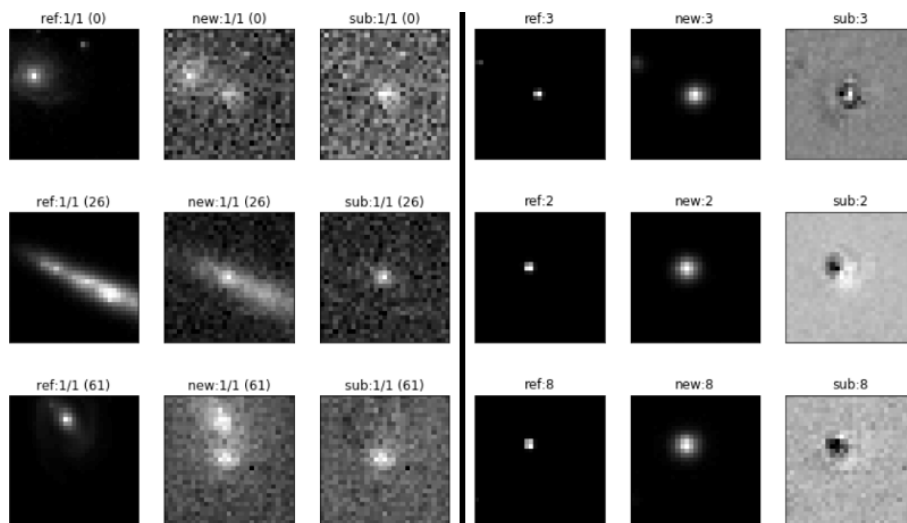


図26 Tomo-e の視野に入り、天体として検出された TNS Transient と、学習データ作成時の Artifact の画像の例。左側 3 列が TNS Transient の例で、右側 3 列が Artifact の例。(左の列から Reference 画像、New 画像、Subtracted 画像)

おおよその seeing サイズ) 以内に検出された Tomo-e source_sub の天体である。BOGUS のデータについては、Model の作成時と同様の方法で作成した Artifact を用いる。ただし、この BOGUS のデータについては、突発天体による汚染が含まれている可能性がある。TNS transient(REAL) のデータ数について、2019/09/01 から 2019/10/31 までに報告されている天体のうち Tomo-e で検出された、105 天体を用いる。重複した検出を含めると、データ数は 455 である。Artifact(BOGUS) のデータ数は、6500 である。

6.2 CNN Model の有無による TPR と FPR の変化

	True Positive Rate	False Positive Rate
初期処理のみ	98.6%	85.2%
初期処理 + CNN Model	82.0%	3.4%

表3 TNS transient(REAL) と Artifact(BOGUS) をどれだけ検出するかを表す。

表 3 に初期処理と CNN Model によって TNS transient(REAL) と Artifact(BOGUS) をどれだけ検出するかを示す。True Positive Rate(TPR) は、REAL をどれだけ検出できたかを割合で示しており、False Positive Rate(FPR) は BOGUS をどれだけ誤検出したかを割合で示している。初期処理のみの場合、REAL は 455 天体中 449 天体を検出できた。検出できなかった 6 天体については、図 27 の画像のように reference 画像で天体が、飽和している天体の非常に近い距離に写っているため、飽和状態の判定を行う処理によって検出されなかった。次に、初期処理と CNN Model によって検出された天体について、TPR は 82.0%(373/455) の検出率であり、FPR は 3.4%(220/6500) の検出率であった。

この分類では、評価に用いる指標の節で述べた *CNN probability* の *Threshold* は 0.5 に設定されている。図 28 に初期処理を通過した REAL と BOGUS に対する CNN Model の分類結果を示す。初期処理を通過

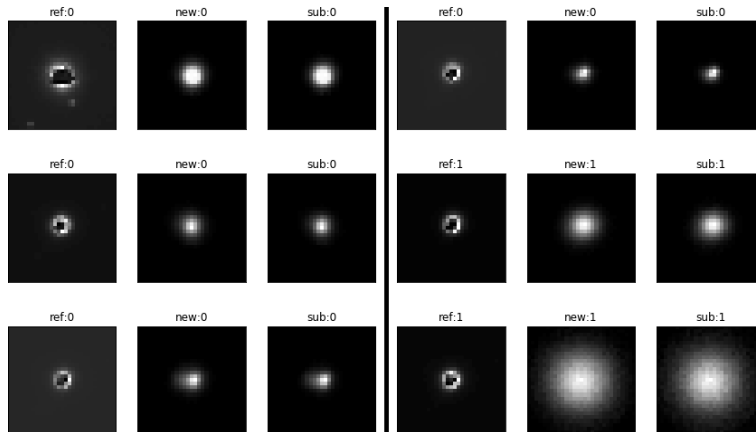


図27 全 REALのうち初期処理により除外された TNS Transient の天体画像を示す。reference 画像で飽和しているため（もしくは飽和している天体の非常に近い距離に天体が写っているため）、初期処理によって Tomo-e Transient の候補から除外された。

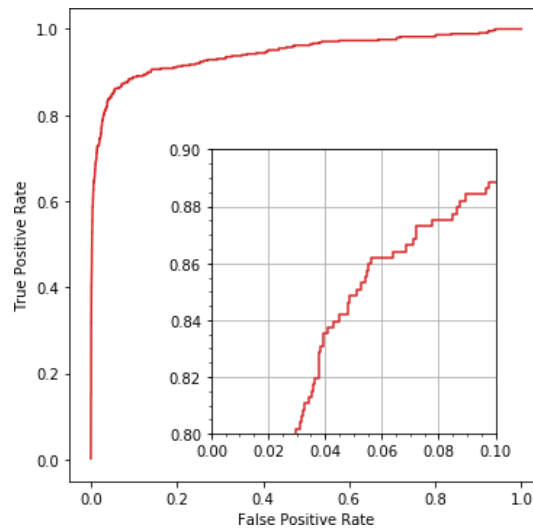


図28 初期処理を通過した REAL と BOGUS に対して CNN Model が分類を行った結果。CNN probability の threshold による分類結果の依存性を表す。曲線下の面積 (AUC) は 0.9432 であった。

した REAL と BOGUS の数は、それぞれ 449、5582 である。横軸は FPR であり、縦軸は TPR である。そのため、曲線が $TPR, FPR = (1, 0)$ に近づくほど（曲線下の面積が大きいほど）性能が良い。図中の赤色の曲線は CNN probability の threshold を 0 から 1 まで推移させたときの分類性能の依存性を示している。図中のサブセットは $0.8 < TPR < 0.9$ 、 $0.0 < FPR < 0.1$ の領域を拡大した図である。AUC の損失について、FPR よりも TPR による損失の方が大きい。そこで、BOGUS であると誤分類した REAL (False Negative) に注目する。CNN probability の threshold が 0.5 のときの Model のパフォーマンスについて、 $TPR = 83.1\%$ 、 $FPR = 96.1\%$ であった。そのときの False Negative の数は 76 である。

6.3 学習データとテストデータに紛れた天体について

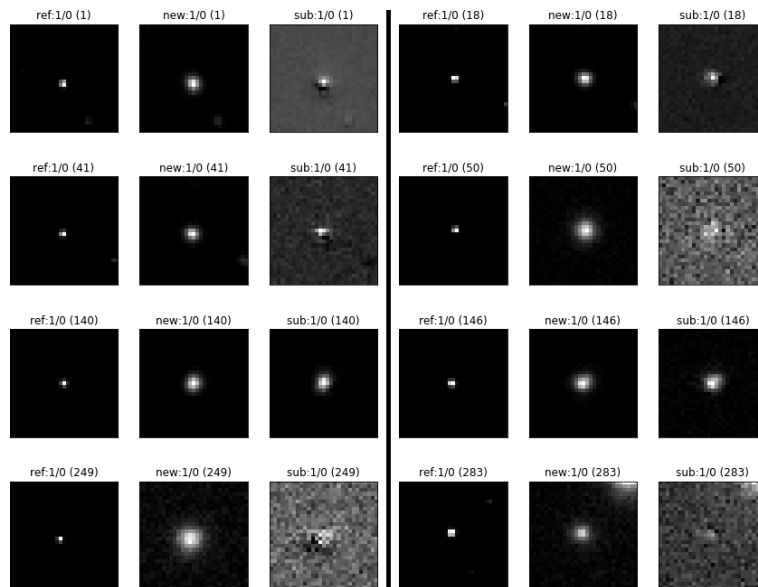


図29 BOGUSであると誤分類した REAL(False Negative) の例。これらは、変光星である可能性が高い。

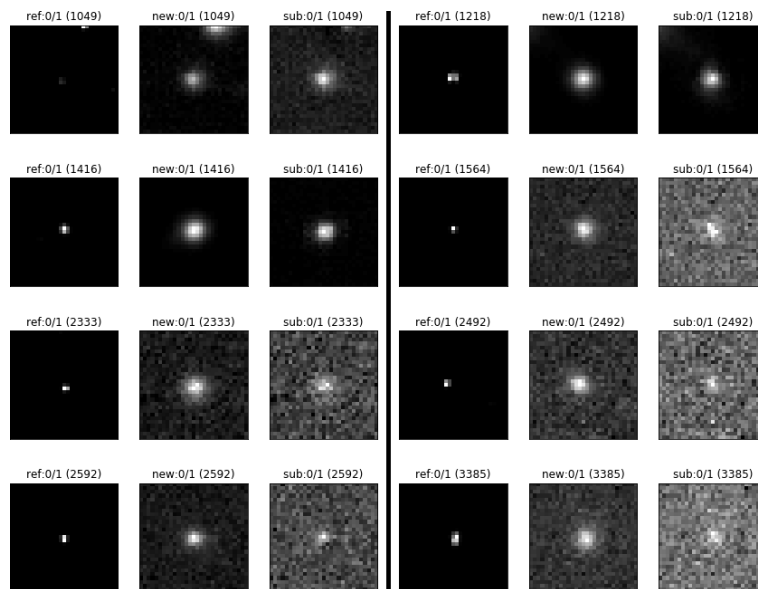


図30 REALであると誤分類した BOGUS(False Positive) の例。これらは、図 30同様にすべて変光星である可能性が高い。

図 29に False Negative の天体画像の一例を示す。False Negative の天体の多くは、transient 天体として報告された座標にすでに reference 画像で写っている天体が多く占めていた。そこでそれらの天体がすでに

Pan-STARRS の天体カタログに登録されているか調べたところ、False Negative の 76 天体の内、53 天体がすでに星として登録されていた。これより、これらの天体は見かけの明るさが変動する変光星である可能性が高い。

図 30 に REAL であると誤分類した BOGUS(False Positive) の天体画像の一例を示す。これらは、図 29 の False Negative の天体画像に酷似している。これら False Positive の 220 天体中 165 天体が Pan-STARRS の天体カタログにてその座標は星として登録されており変光星の可能性が高く、Artifact 作成の際の BOGUS の学習データの汚染となっていると考えられる。

そこで、これら変光星を除いた天体について CNN の Model の分類性能の評価を行う。

テストデータのうち、初期処理を通過し、かつ Pan-STARRS の天体カタログに星として登録されていない天体について、CNN Model の分類パフォーマンスの評価を行う。分類を行う REAL と BOGUS の数はそれぞれ 364, 1315 である。

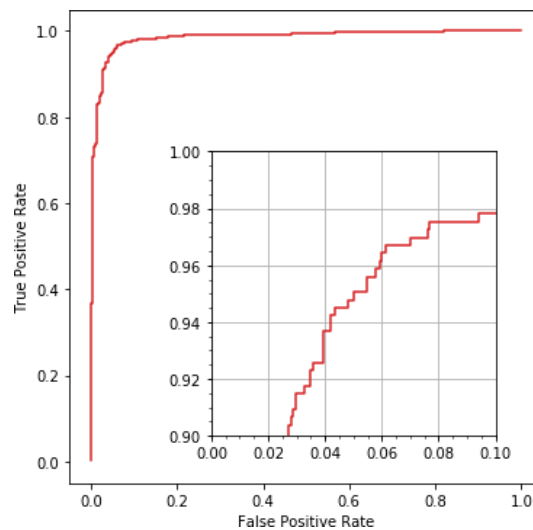


図31 初期処理を通過し、かつ Pan-STARRS の天体カタログに星として登録されていない REAL と BOGUS に対して CNN Model が分類を行った結果。AUC は 0.9844 であった。

図 31 は、図 28 と同様の CNN Model の分類結果を示す。Pan-STARRS の天体カタログに星として登録されている天体を除外することで、AUC は、0.9432 から 0.9844 に改善した。これは、REAL と BOGUS 内に含まれる変光星を除外したため、AUC が改善した。すなわち、学習データ作成時に変光星を BOGUS として分類するよう学習させたためにテストデータの REAL に含まれた変光星を CNN Model が REAL として分類しなかったと考えられる。

6.4 CNN Probability の Threshold の変化に対する性能の依存性

次に、図 31 の TPR と FPR の推移を $Threshold$ を用いて表す。

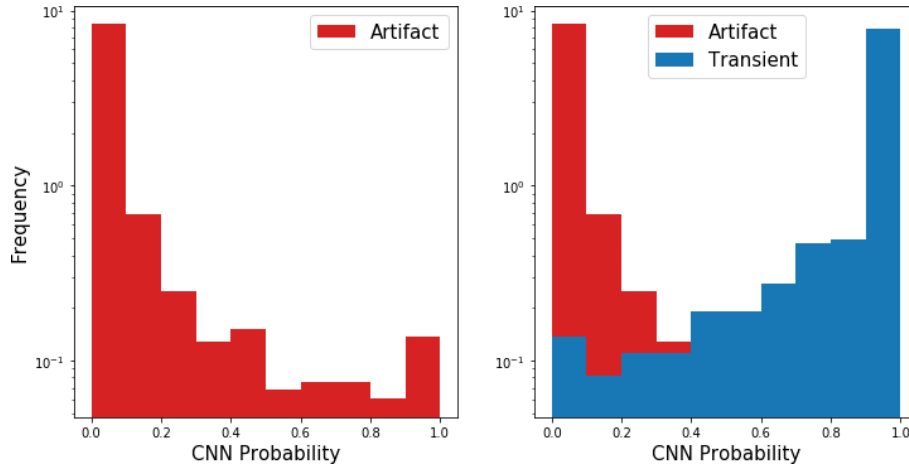


図32 初期処理を通過し、かつ Pan-STARRS の天体カタログに星として登録されていない REAL(Transient) と BOGUS(Artifact) の CNN Probability の度数分布を表す。

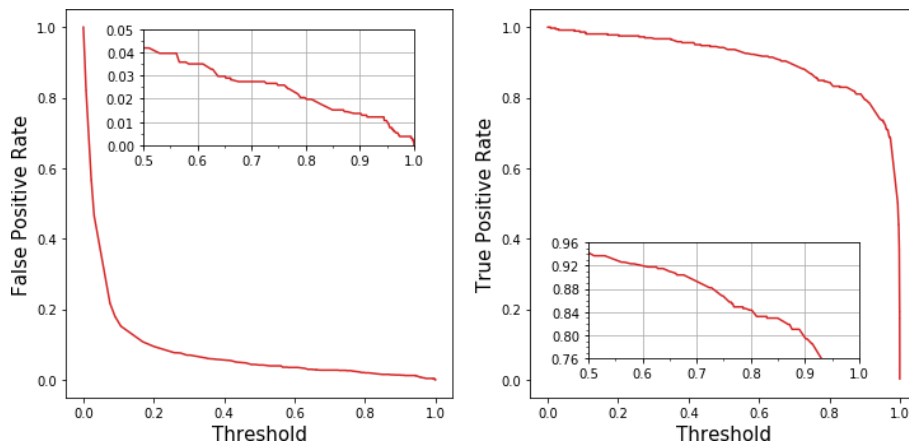


図33 図 31 の結果について、CNN Threshold を用いて TPR と FPR の推移を表す。

図 32 は CNN の Model が評価した天体の CNN Probability の度数分布を表す。CNN Probability はその天体の REAL である確率を表す。とり得る値の範囲は 0 から 1 であり、1 に近いほど REAL である確率が高い。縦軸は Frequency (度数) であり全面積が 1 となるように規格化されている。REAL と BOGUS とも天体の CNN Probability はそれぞれ 1 付近、0 付近に集中していることがわかる。

図 33 は、図 31 の結果を媒介変数である CNN Threshold を用いて TPR と FPR の推移を表している。例えば、Threshold = 0.8 であれば天体の CNN Probability が 0.8 未満であれば BOGUS、それ以上であれば REAL と分類される。Threshold = 0.5 のとき $TPR = 93.7\%$ 、 $FPR = 4.2\%$ だが、Threshold を 0.8 にすれば TPR は 84% に低下するが $FPR = 2.0\%$ と $FPR = 4.2\%$ から検出される BOGUS の数を約半分まで

減らすことが可能である。

6.5 Model の SNR 依存性

この節では天体の差分画像における SNR についての結果と考察を示す。

ROC curve (AUC = 0.9946 / 0.9785)

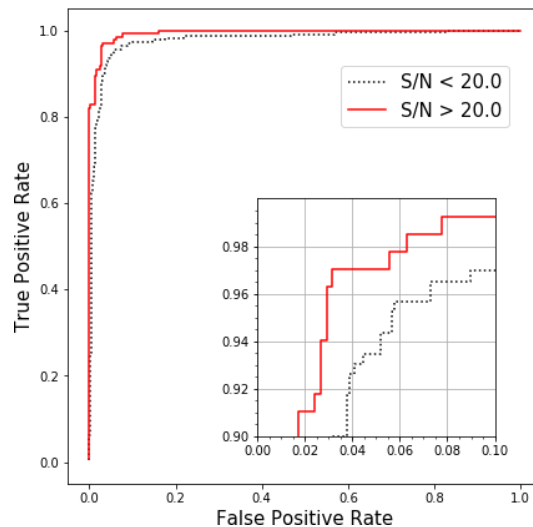


図34 SNR threshold = 20 で、それ以上の天体の ROC curve (赤色実線) と、それ未満の天体の ROC curve (黒色破線) を示す。図 24 と同様に SNR > 20 の天体の方が AUC のスコアは高く、SNR < 20 の ROC curve を上回っていることがわかる。

前章の REAL のデータが Artificial Real からなるテストデータでの評価と同様に、SNR に Threshold を設けてそれ以上もしくはそれ未満の天体に対する Model の分類性能の比較を行う。図 34 は SNR < 20 と SNR > 20 のデータの ROC curve を表す。図中赤色実線の SNR > 20 のデータの方が、破線の SNR < 20 のデータよりも AUC が高い。前章の評価の結果と同様に、天体の像が明瞭であると Model の性能が高くなる。

6.5.1 SNR による AUC の変化

Model の分類性能が SNR に依存しているかを調べるために SNR に Threshold を設け、それ以上もしくはそれ未満の天体に対する Model の分類性能の評価をそれぞれ行う。図 35 について、左図は SNR の Threshold を変化させたとき、その Threshold 以上のデータの AUC ($SNR > SNR \text{ threshold}$) とそれ未満のデータの AUC ($SNR < SNR \text{ threshold}$) がどのように変化するかを表す。

右図は Threshold を変化させた時、Threshold 以上のデータ数 $N(SNR > SNR \text{ threshold})$ の、全体のデータ数に対する割合 $N(SNR > SNR \text{ threshold})/N_{all}$ がどのように変化するかを示しており、Threshold 以上のデータ数の累積度数分布である。青色の曲線は REAL(Transient) のデータ数の割合の推移を示しており、赤色の曲線は BOGUS(Artifact) のデータ数の割合の推移を示している。

SNR について、3 段目右図について、REAL と BOGUS ともに SNR が良い天体ほど数が少ない。また、Elongation と比べて、BOGUS よりも REAL の方が高い SNR を占める割合が大きい。3 段目左図について、Threshold = 20 付近で AUC ($SNR > SNR \text{ threshold}$) のスコアが 0.01 ポイントほど上昇している。これ

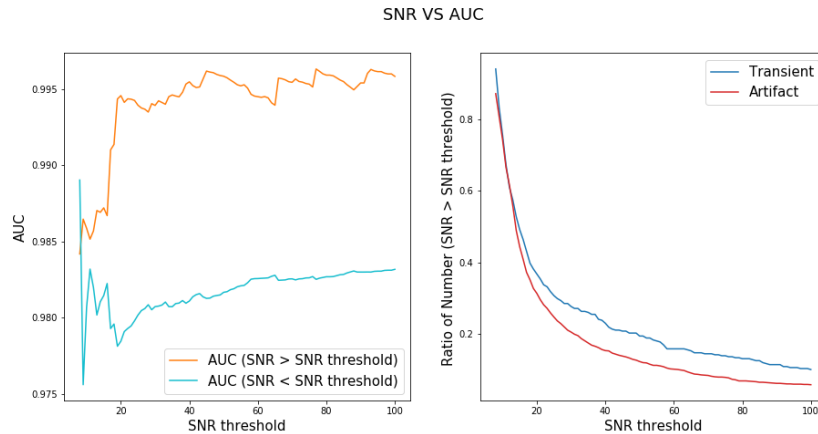


図35 SNR に $threshold$ を設け、それ以上もしくはそれ未満の天体に対する分類性能の評価をそれぞれ行う。右図は $threshold$ を変化させた時の $threshold$ よりも大きい値を持つ天体の数 $N(SNR > SNR threshold)$ の割合を表す。左図は $SNR > SNR threshold$ 、 $SNR < SNR threshold$ それぞれの分類性能を AUC で評価している。詳細は本文に記載。

は、 $SNR = 20$ 付近よりも明瞭に写る天体の分類性能がそれ以下と比べて高いことを示している。その値を超える REAL の数はおよそ 4 割ほどであり、かつ、BOGUS の割合は 3 割と、それよりも少ない。この傾向は $SNR threshold$ が上昇しても変わらず、より厳しい精度が要求されるアラートシステムの構築の際に有効に活用できる。

6.5.2 SNR の閾値による検出数の変化

次に、自動アラートシステムに向けて SNR に Threshold をかけてどれだけの数が検出されるか見積もる。なお、天体の REAL、BOGUS の判定基準である $CNN Probability$ の $Threshold$ は 0.5 で評価を行う。

上述の分析に用いたデータは、2019 年 9 月-10 月の間に Tomo-e で検出されたデータである。このデータは本稿で述べた CNN Model を実装する以前のデータであり、この Model が実際にどれだけの数を検出するかは確認できていない。そこで、この Model を実装後に取得した 2019 年 12 月のデータを用いて検出数の評価を行う。

Positive (True Positive + False Positive) の数の計測には 2019 年 12 月のデータを用いる。ただし、このうち True Positive の数の見積もりには 2019 年 9 月-10 月に得られたデータを用いる。

図 36 は、 $SNR Threshold$ を変化させたときの、CNN model が REAL と分類したデータ数の変化と、そのときの $Precision$ を表す。

$$Precision = \frac{TruePositive}{TruePositive + FalsePositive} \quad (13)$$

$Precision$ は、式 13 で表され、Positive と判定された天体のうち REAL の天体を検出できる割合を示している。例えば 50% であれば 100 検出したもののうち 50 は REAL がであるという目安になる。

左図の縦軸 $Number$ は一晩で検出される天体の数が $SNR Threshold$ を設けることでどれだけ変化するかを示す。”一晩”とは観測時間が 8 時間であり、1exposure に 12 秒かかり、1exposure で Tomo-e Q1 - Q4 のモジュールすべて稼働していることを前提に 84 枚撮影されるとする。そのとき一晩で撮影される画像は

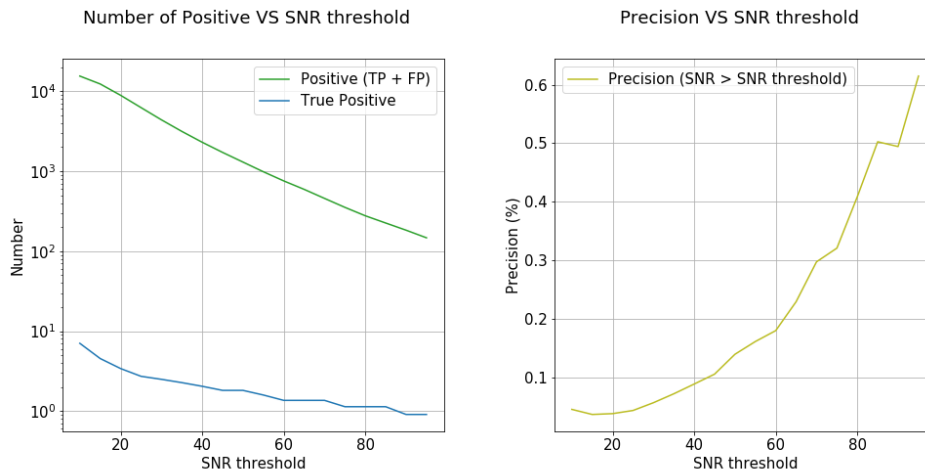


図36 左図： SNR Threshold に対して CNN model が real と分類した (Positive) のデータ数の変化を示す。青色の曲線は、2019年9月-10月に見つかった TNS Transient の数を元に算出した1日あたりに見つかる REAL の新規 Transient の検出数。緑色の曲線は、2019年12月の全新規 Transient のデータ数を元に算出した1日あたりに見つかる BOGUS の新規 Transient の検出数と青色の曲線のデータ数を足したもの。右図：左図のデータを元に SNR Threshold を変化させたときの Precision の推移を示す。 SNR Threshold = 80 のとき Precision はおよそ 0.4 であり、300 回の検出のうち 1.2 回 REAL が検出できる可能性がある。

201,600 枚である。また、2019年9月-10月の取得した画像の枚数は 1,772,968 枚、2019年12月から取得した画像の枚数は 620,334 枚である。

左図の一晩に検出される数 $Number$ のうち True Positive は、2019年9月-10月に検出された True Positive の天体の SNR から、 SNR Threshold を変えたとき検出される数に $201600/1772968$ を掛けて算出した。Positive は 2019年12月に検出された天体の SNR から、 SNR Threshold を変えたときに検出される数に $201600/620334$ を掛けて算出した。

右図から分かるように SNR Threshold を厳しく設けるほど Positive の数は指数関数的に減少する。左図は、式 13 によって計算した Precision である。 SNR Threshold = 80 のとき、Precision はおよそ 0.4% であり、300 回の検出のうち 1.2 回 REAL が検出できる可能性がある。もちろん Threshold をもっと厳しく設定するとその分 Precision は改善するが、検出される REAL の数自体も減少することに注意しなければならない。

6.5.3 CNN Probability の閾値による検出数の変化

最後に、前節と同様に CNN Probability に Threshold をかけてどれだけの数が検出されるか見積もる。

図 37 は、Probability Threshold を変化させたときの、CNN model が REAL と分類したデータ数の変化と、そのときの Precision を表す。前節の SNR の閾値による検出数の変化での議論と同様に評価を行う。

左図の縦軸 $Number$ は一晩で検出される天体の数が Probability Threshold を設けることでどれだけ変化するかを示す。左図の一晩に検出される数 $Number$ のうち True Positive は、2019年9月-10月に検出された True Positive の天体の Probability から、Probability Threshold を変えたとき検出される数に $201600/1772968$ を掛けて算出した。Positive は 2019年12月に検出された天体の Probability から、

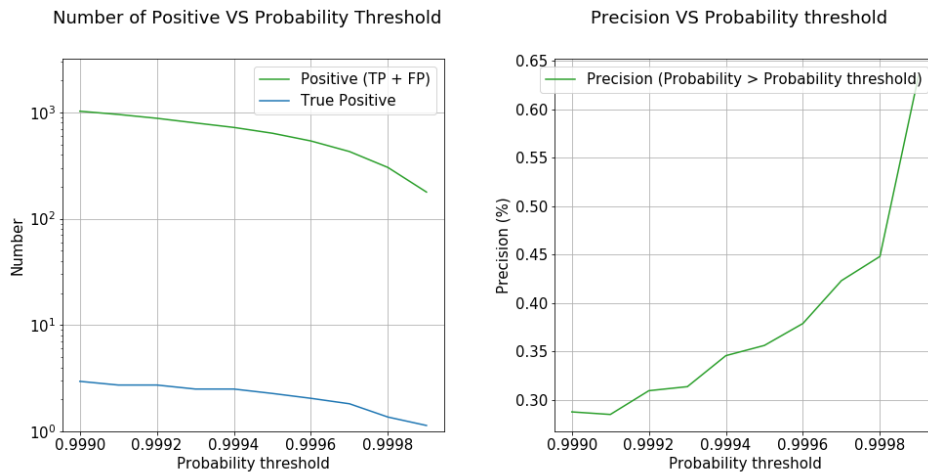


図37 左図：*Probability Threshold* に対して CNN model が real と分類した (Positive) のデータ数の変化を示す。前述の図と同様の構成で、青色の曲線は、2019年9月-10月に見つかった TNS Transient の数を元に算出した1日あたりに見つかる REAL の新規 Transient の検出数。緑色の曲線は、2019年12月の全新規 Transient のデータ数を元に算出した1日あたりに見つかる BOGUS の新規 Transient の検出数と青色の曲線のデータ数を足したもの。右図：左図のデータを元に *Probability Threshold* を変化させたときの *Precision* の推移を示す。*Probability Threshold* = 0.998 のとき *Precision* はおよそ 0.45 であり、300 回の検出のうち 1.3 回 REAL が検出できる可能性がある。

Probability Threshold を変えたときに検出される数に 201600/620334 を掛けて算出した。

右図から分かるように *Probability Threshold* を厳しく設けるほど *Positive* の数は減少する。左図は、式 13 によって計算した *Precision* である。*Probability Threshold* = 0.998 のとき、*Precision* はおよそ 0.45% であり、300 回の検出のうち 1.3 回 REAL が検出できる可能性がある。もちろん *Threshold* をもっと厳しく設定するとその分 *Precision* は改善するが、前節の *SNR* の閾値による検出数の変化での議論と同様に、検出される REAL の数自体も減少することに注意しなければならない。

7 まとめ

Tomo-e Supernova Survey では、1 晩で $10^5 \sim 10^6$ の天体が検出される。しかしそのほとんどが差分処理に失敗した Artifact 天体であり、新しく見つかる Transient 天体は 1 晩で ~ 10 天体ほどと見積もられている。Artifact 天体に埋もれた Transient 天体を効率よく探すために、画像認識の手法である畳み込みニューラルネットワークを用いて、人の目による Transient 探査をサポートできる Model を作成した。Model の作成には、実際に写る星の像から作った Artificial Real 天体を REAL の学習データとして用いた。作成した Model の性能を評価したところ、この Model によって一晩に検出される Transient 候補天体の数は約 15,000 まで絞ることに成功した。

また、Model の性能は信号対雑音比 (SNR) に依存しており、およそ $SNR > 20$ の天体はそれ以下の天体に比べて分類の性能が高いことが分かった。そこで 1 晩の検出の精度 $Precision$ を調べたところ、 $SNR > 80$ の天体については、 $Precision = 0.4\%$ であり、全 Transient 候補天体 300 天体中本物の Transient 天体が 1.2 つ見つかる性能であった。Tomo-e Supernova Survey では自動アラートも視野に入れている。自動アラートで求められる $Precision$ は数十% であり、そのためには今よりも 100 倍ほど性能を上げなくてはならない。自動アラートに用いる Model 作成には画像による特徴量の抽出だけでなく母銀河との距離や超新星の発生場所など、より幅広い特徴量を用いることが必要であると考えられる。

謝辞

本研究を進めるにあたり、東北大学大学院理学研究科天文学専攻の田中雅臣准教授、東京大学大学院理学系研究科附属天文学教育研究センターの諸隈智貴助教には数多くの助言や議論、協力をいただきましたことを深く御礼申し上げます。また、研究会などの様々な場で議論をいただいた Tomo-e Gozen に関わる皆様には活発な議論をいただき大変お世話になりました。そして、日ごろの研究活動を行う中で、多大なご指導を頂いた甲南大学工学部の須佐元教授、指導教員の富永望教授そして、理論研究室の皆様には大変お世話になりましたことを感謝申し上げます。

参考文献

- [1] Ackley, K., Eikenberry, S., S., et al. 2019, "Automated transient detection with shapelet analysis in image-subtracted data", *AJ*, 158, 172
- [2] Alard, C., & Lupton, R. H. 1998, "A Method for Optimal Image Subtraction.", *ApJ*, 503, 325
- [3] Artola, R., Beroiz, M., Cabral, J. 2020, "TOROS Optical follow-up of the Advanced LIGO-VIRGO O2 second observational campaign", *MNRAS*, stz3634
- [4] Bersten, M. C., Folatelli, G., García, F., et al. 2018, "A surge of light at the birth of a supernova", *Nature*, 554, 497
- [5] Bertin, E., & Arnouts, S. 1996, "SExtractor: Software for source extraction", *A&AS*, 117, 393
- [6] Brink, H., Richards, J. W., Poznanski, D., et al. 2013, "Using Machine Learning for Discovery in Synoptic Survey Imaging", *MNRAS*, 435, 1047
- [7] Cabrera-Vives, G., Reyes, I., Förster, F., Estévez, P. A., & Maureira, J. 2012, "Deep-HiTS: Rotation Invariant Convolutional Neural Network for Transient Detection", *ApJ*, 836, 97
- [8] Duev, D. A., Mahabal, A., Masci, F. J., et al. 2019, "Real-bogus classification for the Zwicky Transient Facility using deep learning", *MNRAS*, 489, 3582
- [9] Filippenko, A. V. 1997, "Optical Spectra of Supernovae", *ARA&A*, 35, 309
- [10] Gieseke F., et al. 2017, "Convolutional Neural Networks for Transient Candidate Vetting in Large-Scale Surveys", *MNRAS*, 472, 3101
- [11] Mahabal, A., Rebbapragada, U., Walters, R., et al. 2019, "Machine Learning for the Zwicky Transient Facility", *PASP*, 131, 038002
- [12] Morii, M., Ikeda, S., Tominaga, N., et al. 2016, "Machine-learning Selection of Optical Transients in Subaru/Hyper Suprime-Cam Survey", *PASJ*, 68, 104
- [13] Sánchez, B., et al. 2019, "Machine Learning on Difference Image Analysis: A comparison of methods for transient detection", *A&C*, 28, 100284
- [14] Sedaghat, N., Mahabal, A. 2018, "Effective Image Differencing with ConvNets for Real-time Transient Hunting.", *MNRAS*, 476, 5365
- [15] Selvaraju, R. R., Das, A., Vedantam, R., Cogswell, M., Parikh, D., Batra, D. 2016, "Grad-cam: Why did you say that? visual explanations from deep networks via gradient-based localization.", *arXiv:1611.01646*
- [16] Simonyan, K., Vedaldi, A., Zisserman, A., 2013, "Deep inside convolutional networks: Visualising image classification models and saliency maps.", *arXiv:1312.6034*

- [17] Soderberg, A. M., et al. 2008, "An extremely luminous X-ray outburst at the birth of a supernova", *Nature*, 453, 469
- [18] Wright, D. E., Smartt, S. J., Smith, K. W., et al. 2015, "Machine learning for transient discovery in Pan-STARRS1 difference imaging.", *MNRAS*, 449, 451
- [19] Zackay, B., Ofek, E.O., Gal-Yam, A. 2016, "Proper Image Subtraction: Optimal Transient Detection, Photometry, and Hypothesis Testing.", *ApJ*, 830, 27
- [20] 小島悠人, 2019, "木曾超広視野高速 CMOS カメラの性能評価及び高速移動する地球接近天体の広視野探査", 東京大学大学院理学系研究科天文学専攻修士論文
- [21] 山田章一 (著), 2016, "新天文ライブラリー第4巻超新星", 日本評論社
- [22] the Tomo-e Gozen Project, <http://www.ioa.s.u-tokyo.ac.jp/tomoe/index.html>
- [23] 斎藤康毅 (著), 2016, "ゼロから作る Deep Learning Python で学ぶディープラーニングの理論と実装", O'Reilly
- [24] Francois Chollet(著), 株式会社クイープ (訳), 2018, "Python と Keras によるディープラーニング", マイナビ BOOKS

付録 A Machine Learning のスクリプト

Machine Learning の Model 作成に関する Python スクリプトは、
tpweb2.phys.konan-u.ac.jp/hamasaki/tomoesnml.html
に公開している。